

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Les performances et la sécurité d'un réseau local

Donnen, Dany; Delvaux, Alain

Award date:
1990

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

**FACULTES
UNIVERSITAIRES
N.D. DE LA PAIX**

NAMUR

INSTITUT D'INFORMATIQUE

**LES PERFORMANCES
ET LA SECURITE
D'UN RESEAU LOCAL**

**par Alain DELVAUX
et
Dany DONNEN**

**Promoteur :
Professeur J. RAMAEKERS**

Mémoire présenté en vue
de l'obtention du titre
de Licencié et Maître
en Informatique

Année académique 1989 - 1990

Nous tenons à exprimer nos plus vifs remerciements à monsieur le professeur Ramaekers sans l'aide de qui nous n'aurions pu mener à bien ce mémoire.

Nous remercions avec grand plaisir Bernard Detrembleur et Paul-Georges Crismer pour l'aide considérable qu'ils nous ont apportée tout au long de la réalisation de ce travail.

Nous exprimons également toute notre gratitude à Jean-Pierre Hogue, Bruno Delcourt, Gérard Paquet et Jean-Marie Lheureux pour les conseils judicieux qu'ils n'ont pas hésité à nous donner.

Nous y associons Jean-Marc Verdure et Jean-Paul Leclercq pour l'avis éclairé dont ils nous ont fait bénéficier.

Abstract

Security and performances of a local area network are two essentials components of its good running.

The security of the network of the Computer Science Institut has been considered by using a client/server model. This approach allow to evaluate each factor in the process of security of a system and to locate the weaknesses of some servers.

The mesure of the performances of the system has been considered as a help to configuration management. A measurement software of the evolution of the activity has been developped to constitute for the manager a decision help tool.

This two approaches have allowed to show that, both the security level and the health state of the studied system, answer to the needs of the users and the managers.

Condensé

La sécurité et les performances d'un réseau local constituent deux éléments essentiels de son fonctionnement.

La sécurité du réseau local de l'Institut d'Informatique a été envisagée selon un modèle client/serveur. Cette approche permet d'évaluer chaque facteur intervenant dans le processus de sécurisation d'un système et de localiser les failles inhérentes à certains serveurs.

La mesure des performances du système a été considérée comme une aide au management de la configuration. Un logiciel de mesure de l'évolution de l'activité a été développé, dans le but de constituer pour l'administrateur un premier outil d'aide à la décision.

Ces deux approches ont permis de montrer que tant le niveau de sécurité que l'état de santé du système étudié répondent aux besoins des utilisateurs et des administrateurs.

INTRODUCTION	1
---------------------------	----------

CHAPITRE 1: LE PROBLEME ET SON CONTEXTE

1.1. DESCRIPTION DU PROBLEME	4
1.2. CONFIGURATION DU RESEAU DE L'INSTITUT	6
1.2.1. INTRODUCTION.....	6
1.2.2. CONFIGURATION DU RESEAU.....	6
1.2.1. CONCEPTS DE BASE	6
1.2.2. ACCES PAR LE RESEAU.....	7
1.2.3. ACCES VIA LE PABX	9
1.2.3. ARCHITECTURE PHYSIQUE DU RESEAU.....	11

CHAPITRE 2 : LA SECURITE

2.1. APPROCHE CONCEPTUELLE ET OBJECTIFS.....	16
2.1.1. INTRODUCTION.....	16
2.1.2. DEFINITION DES CONCEPTS.....	17
2.1.2.1. LES NIVEAUX DE SECURITE.....	17
2.1.2.2. LES TYPES DE RISQUES.....	18
2.1.2.3. LES OBJECTIFS DE LA SECURITE	19
2.1.3. OBJECTIFS.....	19
2.2. MODELE ET METHODOLOGIE.....	21
2.3. ANALYSE DES RISQUES ET DES BESOINS.....	23
2.3.1. LES FACTEURS DE RISQUE.....	23

2.3.2. TYPOLOGIE DES RISQUES.....	23
2.3.2.1. LES ERREURS.....	23
2.3.2.1.1. Les erreurs de saisie.....	23
2.3.2.1.2. Les erreurs d'exploitation.....	24
2.3.2.1.3. Les erreurs de conception.....	24
2.3.2.2. LES FRAUDES ET ACTES VOLONTAIRES	25
2.3.2.2.1. Les menaces.....	25
2.3.2.2.2. Les techniques d'attaque	27
2.3.3. EVALUATION DES RISQUES	32
2.3.3.1. LES ERREURS	32
2.3.3.1.1. Les erreurs indirectes.....	33
2.3.3.1.2. Les erreurs directes	37
2.3.3.2. LES ACTES VOLONTAIRES	38
2.3.3.2.1. Les attaques passives.....	39
2.3.3.2.2. Les attaques actives	40
2.3.4. EVALUATION DES BESOINS	44
2.3.4.1. LES BESOINS DES UTILISATEURS.....	44
2.3.4.2. LES BESOINS DE L'ADMINISTRATEUR	45
2.4. SERVICES ET SERVEURS DE SECURITE.....	46
2.4.1. LE SYSTEME D'EXPLOITATION : UNIX.....	46
2.4.1.1. SERVICES AUX UTILISATEURS	46
2.4.1.1.1. Le mot de passe.....	46
2.4.1.1.2. Les groupes	47

2.4.1.1.3.	La protection des fichiers	48
2.4.1.1.4.	La protection des répertoires.....	49
2.4.1.1.5.	La modification des protections	51
2.4.1.1.6.	Les protections par défaut.....	51
2.4.1.1.7.	Les SUID et SGID.....	52
2.4.1.1.8.	Le chiffrement des fichiers	53
2.4.1.2.	SERVICES A L'ADMINISTRATEUR.....	55
2.4.1.2.1.	Les fichiers et répertoires système.....	55
2.4.1.2.2.	La gestion des utilisateurs.....	59
2.4.1.2.3.	Les commande système	59
2.4.2.	LES UTILISATEURS.....	60
2.4.2.1.	LA GESTION DU MOT DE PASSE.....	60
2.4.2.1.1.	Choix du mot de passe.....	60
2.4.2.1.2.	Durée de validité du mot de passe.....	62
2.4.2.2.	LA PROTECTION DES DONNEES	62
2.4.2.2.1.	La protection des fichiers et répertoires.....	62
2.4.2.2.2.	Le chiffrement des fichiers	65
2.4.3.	L'ADMINISTRATEUR.....	66
2.4.3.1.	LE CONTROLE EXTERNE.....	66
2.4.3.2.	LE CONTROLE INTERNE.....	68
2.4.3.3.	LA FORMATION DES UTILISATEURS	71
2.4.4.	LES OUTILS LOGICIELS.....	74
2.5.	CONCLUSIONS.....	77

CHAPITRE 3 : LES PERFORMANCES

3.1. INTRODUCTION.....	81
3.2. ANALYSE DU PROBLEME.....	83
3.2.1. EVALUATION DES BESOINS	83
3.2.1.1. LES BESOINS GLOBAUX.....	83
3.2.1.1.1. Le temps de réponse.....	83
3.2.1.1.2. Le pourcentage de disponibilité	86
3.2.1.1.3. Le volume de transactions	86
3.2.1.1.4. La charge des lignes de communication.....	87
3.2.1.1.5. Le délai moyen de transfert.....	88
3.2.1.1.6. Le pourcentage de travaux exécutés à temps	89
3.2.1.1.7. Le pourcentage de réexécution.....	89
3.2.1.1.8. L'utilisation des machines	90
3.2.1.2. LES BESOINS SPECIFIQUES.....	90
3.2.2. ANALYSE DES OUTILS DE PRISE DE MESURES.....	93
3.2.2.1. OUTILS GENERALEMENT UTILISES.....	93
3.2.2.1.1. Les outils de prise de mesures.....	93
3.2.2.1.2. Les outils de présentation	97
3.2.2.2. OUTILS DISPONIBLES.....	99
3.2.2.2.1. DxTopCPU.....	100
3.2.2.2.2. Monitor	103
3.2.3. CHOIX D'UNE SOLUTION	108
3.3. DEVELOPPEMENT DE L'OUTIL DE MESURE.....	111
3.3.1. PHILOSOPHIE DE L'OUTIL.....	111
3.3.2. CAHIER DES CHARGES	113

3.3.2.1. MODULE DE PRISE DE MESURES.....	113
3.3.2.1.1. Données collectées.....	113
3.3.2.2. Stockage des résultats.....	115
3.3.2.3. Fréquence des mesures	115
3.3.2.2. MODULE DE PRESENTATION DES RESULTATS.....	115
3.3.2.2.1. Sélection des résultats.....	115
3.3.2.2.2. Sélection du type de présentation.....	116
3.3.2.2.3. Réutilisation des résultats	116
3.3.3. DEVELOPPEMENT	117
3.3.3.1. CHOIX DE DEPART.....	118
3.3.3.1.1. Module de prise de mesures.....	118
3.3.3.1.2. Module de présentation.....	118
3.3.3.2. DEVELOPPEMENT DES MODULES	121
3.3.3.2.1. Module de prise de mesures.....	121
3.3.3.2.1 Module de présentation.....	164
3.4. UTILISATION DU LOGICIEL.....	171
3.4.1. INTRODUCTION.....	171
3.4.2. MANUEL D'UTILISATION.....	172
3.4.2.1. LANCEMENT DU LOGICIEL.....	172
3.4.2.2. SELECTION DES DONNEES	174
3.4.2.2.1. Sélection de la période.....	174
3.4.2.2.2. Sélection des données.....	175
3.4.2.2.3. Sélection de l'hôte.....	179
3.4.2.3. CONSTRUCTION DES TABLEAUX ET GRAPHIQUES.	181

3.5. UTILISATION DE L'OUTIL.....	188
3.5.1 PERTINECE DES VALEURS ET MESURES.....	188
3.5.1.1. CORRECTION DES MESURES.....	188
3.5.1.2. PERTINENCE DES MESURES.....	191
3.5.2. REALISATION DE L'ANALYSE.....	191
3.5.2.1. HYPOTHESES.....	191
3.5.2.2. ANALYSE DES MESURES COLLECTEES.....	193
3.5.2.2.1. Utilisation du CPU.....	193
3.5.2.2.2. Accès au disque.....	199
3.5.2.2.3. Occupation de la mémoire	202
3.5.2.2.4. Activité de pagination.....	204
3.5.2.2.5. Contribution au trafic du réseau.....	206
3.5.2.3. ENSEIGNEMENTS.....	211
3.6. CONCLUSIONS.....	213

CONCLUSIONS	215
--------------------------	-----

ANNEXES	218
----------------------	-----

ANNEXE A	GLOSSAIRE
ANNEXE B	PROGRAMME PWADM
ANNEXE C	PROGRAMME SUW
ANNEXE D	COMMANDES UNIX DE MESURES
ANNEXE E	COMMANDE ETHERFIND SUR SUN
ANNEXE F	CODE SOURCE DU MESUREUR
ANNEXE G	FICHIERS WS*.DAT
ANNEXE H	FICHIERS AUX*.DAT
ANNEXE I	FICHIERS INIT*.DAT
ANNEXE J	FICHIERS COMP*.DAT
ANNEXE K	FICHER HEURE.DAT
ANNEXE L	CRONTAB
ANNEXE M	MACRO-COMMANDE EXCEL
ANNEXE N	FICHER PERF.INTERPRET DE DEC

INTRODUCTION

La sécurité et les performances d'un système informatique sont deux domaines qui suscitent de plus en plus d'intérêt, tant de la part des informaticiens que des responsables économiques et financiers des organisations.

La sécurité est une problématique qui semble bénéficier à l'heure actuelle d'une publicité importante. Les récents événements portant atteinte à différents systèmes importants, cumulés à la prise de conscience du problème par les hommes de loi ont engendré un accroissement systématique du nombre d'études consacrées à ce domaine. Le nombre important d'analyses réalisées a introduit une grande diversité des approches du problème et une vision de la sécurité spécifique à chaque cas étudié.

L'analyse des performances d'un système informatique fait partie intégrante de son cycle de vie. Tant lors du choix qu'au moment de son installation, il est procédé à des mesures permettant d'évaluer l'utilisation et les capacités du système. Cet intérêt doit se prolonger alors même que la configuration est en fonctionnement réel, de façon à permettre, au responsable, d'évaluer son efficacité et de disposer de données correctes. Ces informations lui permettront, tout au long de l'évolution du système, de remplir son rôle de décideur ou de conseiller auprès des dirigeants de l'organisation.

Il est donc logique que l'Institut d'Informatique se penche sur le problème de la sécurité et des performances de la configuration d'ordinateurs dont il dispose. Une telle étude représente un sujet idéal de mémoire laissant à ses auteurs une liberté importante dans la méthode d'approche du problème et offrant au lecteur la possibilité d'ajouter à ses connaissances une analyse de la sécurité et des performances réalisée dans un environnement particulier : l'Université.

CHAPITRE 1

LE PROBLEME

1.1. DESCRIPTION DU PROBLEME

L'analyse des performances et de la sécurité d'un réseau, reliant plusieurs ordinateurs hétéroclites, peut paraître audacieuse de prime abord. Il nous semble donc nécessaire de délimiter clairement la portée du mémoire et de déterminer de façon aussi précise que possible les objectifs de cette étude.

Une remarque préliminaire nous semble toutefois utile. Une analyse de ce type est considérablement influencée par l'environnement qui en est l'objet, la configuration qui nous intéresse mérite une attention particulière à deux points de vue.

D'une part il s'agit d'un environnement universitaire, c'est-à-dire supportant des travaux bien particuliers et que l'on peut classer en deux grandes familles assez proches. Les travaux de recherche et de développement, incombants aux professeurs, assistants et chercheurs de l'Institut d'Informatique et les travaux d'étude réalisés par les étudiants. Cette particularité a une importance sur l'utilisation des ressources.

D'autre part, la configuration technique doit être connue du lecteur pour lui permettre d'appréhender plus rapidement les références aux ordinateurs et aux appareils présents sur le réseau. Cette nécessité s'impose également dans un but de compréhension des objectifs tout d'abord, et des décisions de développement ensuite. Nous avons prévu, à cet effet, une partie de ce premier chapitre qui sera entièrement consacrée à la présentation de la configuration du réseau, tel qu'il existe au sein de l'Institut d'Informatique.

La portée exacte du mémoire est difficile à préciser a priori, d'autant que nos connaissances de départ sont assez réduites. Elle doit néanmoins être délimitée suivant deux angles différents, d'un côté la partie de la configuration faisant réellement l'objet de l'analyse et d'un autre côté les aspects précis des deux domaines qui seront développés. Du point de vue de la configuration, nous nous limiterons aux quatre nouvelles stations Vax et à leur serveur, ce choix est orienté par l'intérêt que représentent ces nouvelles acquisitions aux yeux de l'administrateur du système.

En ce qui concerne la délimitation des aspects de la sécurité et des performances, nous nous proposons d'aborder les deux problèmes à traiter indépendamment l'un de l'autre, même si la frontière qui les sépare n'est pas toujours évidente à tracer. Le chapitre 2 prendra donc en considération l'aspect sécurité du réseau, le problème des performances fera, quant à lui, l'objet du chapitre 3. Nous pensons plus judicieux de revenir en détails sur la délimitation propre à chaque sujet dans le chapitre qui leur est respectivement consacré.

Nous voudrions apparenter l'objectif de ce mémoire à celui d'une étude d'audit et d'évaluation. Cet objectif permet de fournir un résultat concret au lecteur qui si il le désire peut en tirer les leçons intéressantes et les appliquer à la fonction qu'il remplit éventuellement dans le cadre de cette configuration. Cette approche concrète nous a permis de mettre en exergue une dualité présente à plusieurs niveaux du développement et essentiellement due aux choix effectués au niveau de la méthode de travail.

La première dualité est présente entre les domaines étudiés de façon totalement distincte, le but étant de fournir à l'administrateur une évaluation et des propositions propres à chaque aspect. Cette vision lui permet de juger les résultats de l'étude individuellement et de s'inspirer plus ou moins des conclusions tirées d'un des domaines, sans pour autant être tenu par celles de l'autre. Elle autorise de plus le lecteur à ne s'attarder qu'au domaine qui l'intéresse, sans pour cela nécessiter de prérequis émanant de l'autre aspect. La complémentarité existe mais n'est nullement une contrainte.

Une seconde dualité se remarque dans le type d'approche propre à chacun des domaines étudiés. L'aspect sécurité du réseau a été envisagé de façon théorique, son approche est basée sur une modélisation ayant, selon nous, le mérite de faciliter sa compréhension et s'adaptant à notre avis très bien à ce sujet. Les performances du système ont été, quant à elles, abordées d'un point de vue pratique pour deux raisons principales. Le souhait de Monsieur le Professeur Ramaekers était de disposer d'un outil d'aide à la décision et à la gestion du réseau de l'Institut, souhait tout à fait légitime et qui nous paraissait réalisable. L'argument supplémentaire en faveur d'une approche pratique est de rentrer parfaitement dans l'objectif global de ce mémoire, l'évaluation d'un système et la proposition de solutions d'aide à sa gestion.

1.2. LA CONFIGURATION DU RESEAU DE L'INSTITUT

1.2.1. INTRODUCTION

Les réseaux connaissent actuellement un développement important, leur capacité de partage de ressources et leur possibilité de communication et d'échange de données en sont les facteurs majeurs. Toutefois leur utilisation nécessite avant tout une connaissance globale du réseau et des machines interconnectées, connaissance qui permettra à l'utilisateur futur de la configuration d'appréhender plus efficacement les possibilités du système.

Les moyens calculs accessibles à toutes les facultés sont gérés par le Centre de Calcul. L'accès à ces services est possible grâce à un réseau reliant les divers bâtiments du site universitaire. L'Institut d'Informatique de par ses besoins spécifiques s'est équipé de diverses machines elles-mêmes reliées à ce réseau.

Cette présentation du réseau se limitera aux accès fournis par le Centre de Calcul à l'Institut d'Informatique ainsi qu'aux moyens propres de ce dernier.

1.2.2. CONFIGURATION DU RESEAU

1.2.2.1. CONCEPTS DE BASE

Un **cluster** est un groupement de machines qui permet à l'utilisateur de les voir comme une seule.

Un **serveur** est une machine qui fournit et qui gère des ressources partagées par d'autres (disque, imprimante, ...).

Un **serveur de terminal** est un appareil qui permet l'accès au réseau à partir d'un terminal.

Une **station de travail** est un ordinateur individuel faisant partie d'un système distribué et répondant à certaines caractéristiques techniques.

Un **protocole** est un ensemble de règles permettant à deux machines de se comprendre. Deux machines ne partageant pas le même protocole ne peuvent donc pas communiquer.

X25 est un protocole d'accès au réseau public DCS de la RTT destiné aux transmissions de données digitales par commutation de paquets.

TCP/IP (Transmission Control Protocole / Internet Protocole) est un exemple très répandu de protocole de communication en réseau local.

DECnet est un protocole de communication appartenant à Digital Equipement Corp.

Un **PABX** est un auto-commutateur privé qui réalise la commutation et le transfert de signaux téléphoniques et digitaux. Il joue un rôle comparable à celui d'une centrale téléphonique.

1.2.2.2. ACCES PAR LE RESEAU

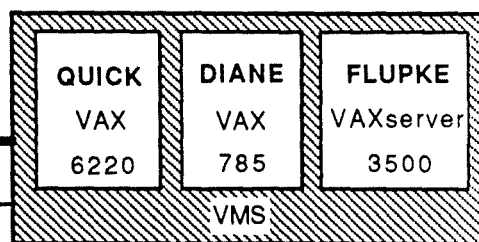
Le Centre de Calcul fournit à l'Institut d'Informatique la possibilité d'accéder à certaines de ses machines situées au premier étage de l'Institut. Ces machines connues sous le nom de QUICK, DIANE et FLUPKE sont organisées en cluster se partageant une unité disque commune. Le Centre de Calcul dispose également d'un serveur de terminaux. Cette structure est illustrée à la figure 1.

L'Institut d'Informatique dispose actuellement de trois serveurs et de quinze stations de travail. Le serveur JUNON bien qu'appartenant à l'Institut, est localisé au Centre de Calcul pour des raisons techniques. Tout comme le cluster il permet l'accès vers l'extérieur selon le protocole X25. Ce serveur supporte le protocole d'accès TCP/IP. Il est clair que JUNON et IDA ne peuvent communiquer directement. Les stations IDA, WS5 et WS6 supportent tout comme le cluster le protocole d'accès DECnet. Le serveur SUNSERV1 ainsi que les stations de travail ULYSSE, PRIAM, MKEE, SKEE, KAOS1 et FOLON1 à FOLON3 travaillent sur TCP/IP. Le serveur ELECTRE et les quatre stations WS1 à WS4, supportant les deux protocoles précités, peuvent communiquer indifféremment avec tous les autres hôtes.

Il est donc possible d'accéder à un serveur à partir de n'importe quelle station de travail supportant le même protocole, sinon ELECTRE peut être utilisée comme "passerelle" pour un accès en cascade. Cette procédure est appelée le **remote login** et fait l'objet d'explications détaillées dans un autre rapport [DET, 16].

CENTRE DE CALCUL

CLUSTER



INSTITUT D'INFORMATIQUE

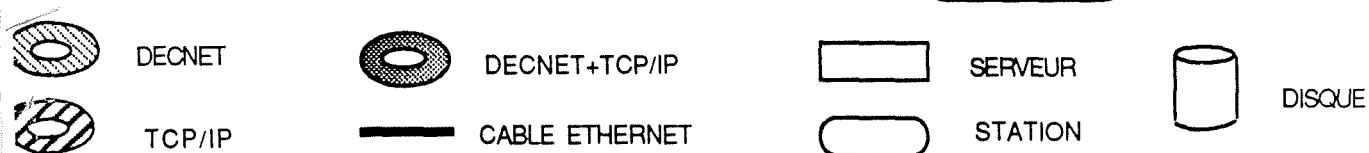
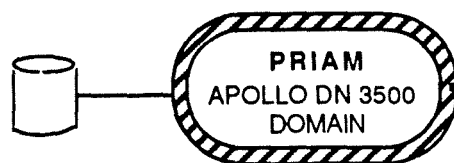
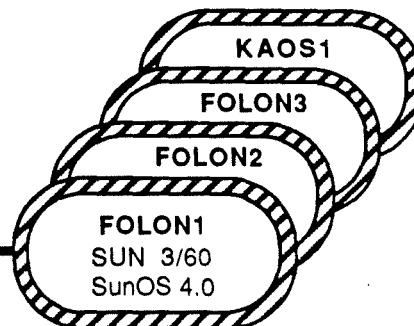
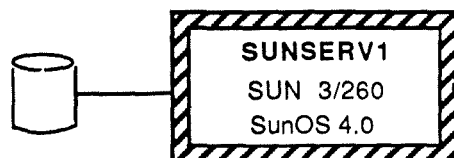
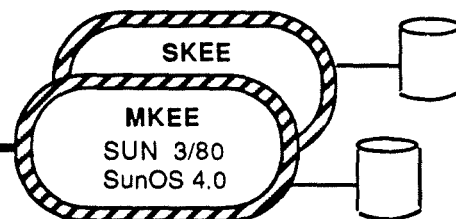
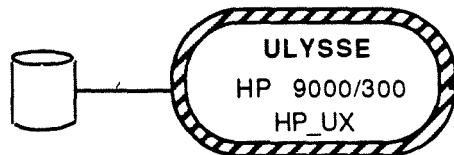
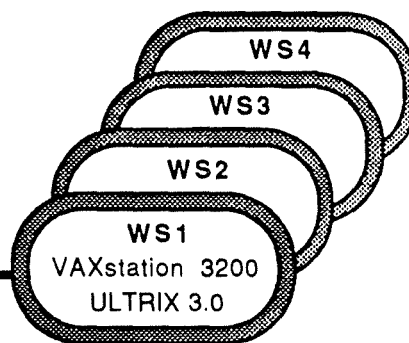
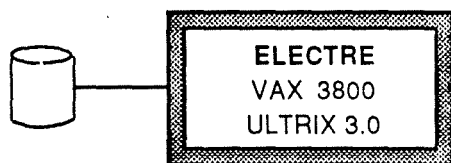
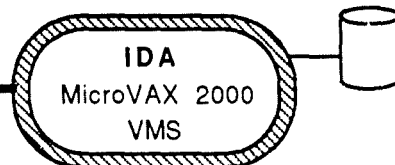
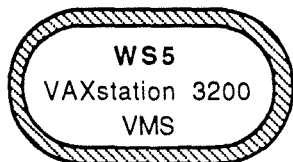
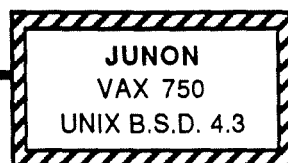
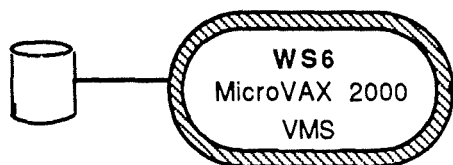


FIG 1 : CONFIGURATION DU RESEAU

1.2.2.3. ACCES VIA LE PABX

Le PABX permet l'accès au réseau à partir d'un terminal en réalisant la commutation vers le serveur de terminaux (en demandant le service srv). Comme le montre la figure 2, il est également possible d'atteindre les serveurs ELECTRE et JUNON par les services de même nom ainsi que SUNSERV1 grâce aux services sun1, 2 et 3.

Le PABX est également connecté par un cable en fibre optique à la centrale téléphonique des Facultés. La possibilité est donc offerte d'atteindre le réseau à partir de l'extérieur en utilisant le réseau téléphonique de la RTT.

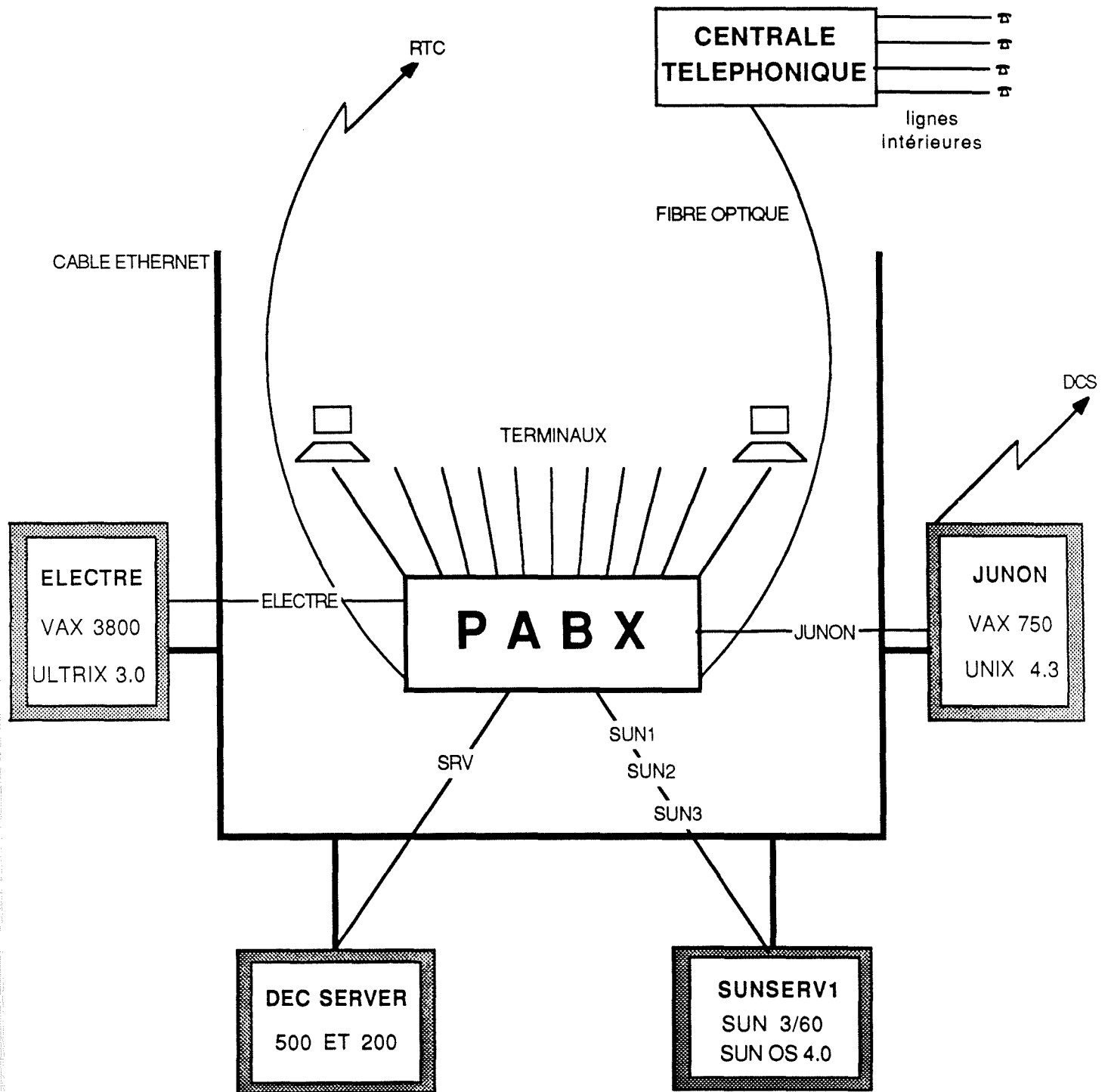


FIG 2 : ACCES VIA LE PABX

1.2.3. ARCHITECTURE PHYSIQUE DU RESEAU

Le réseau des Facultés est basé sur deux câbles coaxiaux principaux appelés **backbone**. La première de ces deux épine dorsales assure la connexion entre l'Institut d'Informatique, la Faculté des Sciences Economiques, la Faculté de Droit et la bibliothèque, la seconde reliant les différents bâtiments de la Faculté des Sciences (à l'exception des Sciences Mathématiques). Ces deux câbles sont le support physique de transmission du réseau local de type ETHERNET.

Un réseau ETHERNET est basé sur la règle de base suivante : à un instant donné, une seule machine "parle" et toutes les autres "écoutent" , si une autre veut "parler", elle doit attendre que toutes les autres "se taisent". La topologie utilisée par le réseau ETHERNET est le **bus** (figure 3).

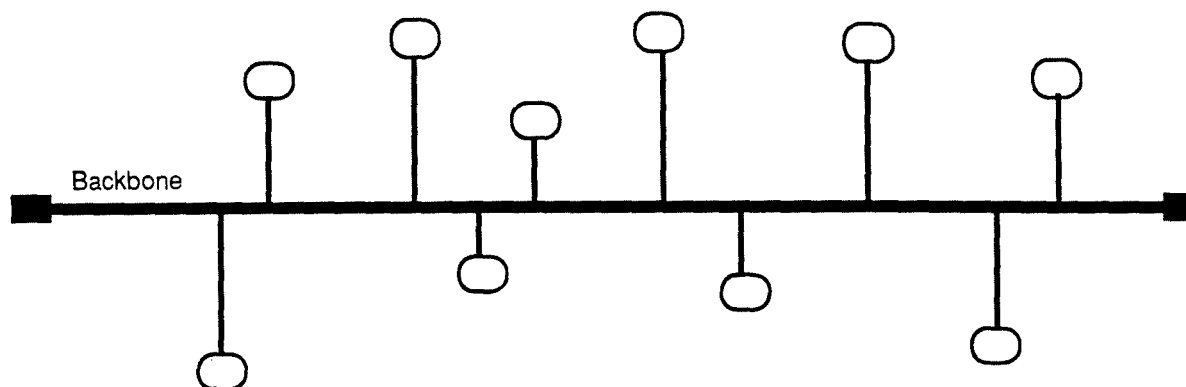


FIG 3 : BUS ETHERNET

Le moyen de transmission est composé d'un ou plusieurs câbles de longueur limitée à 500 m et appelé un **segment**. Un **terminateur** doit être placé à chaque extrémité d'un segment, cet appareil maintenant la tension constante sur le câble.

Comme illustré à la figure 4, les segments peuvent être interconnectés par un système permettant de répéter les signaux d'un segment à l'autre (**remote repeater, bridge, ...**).

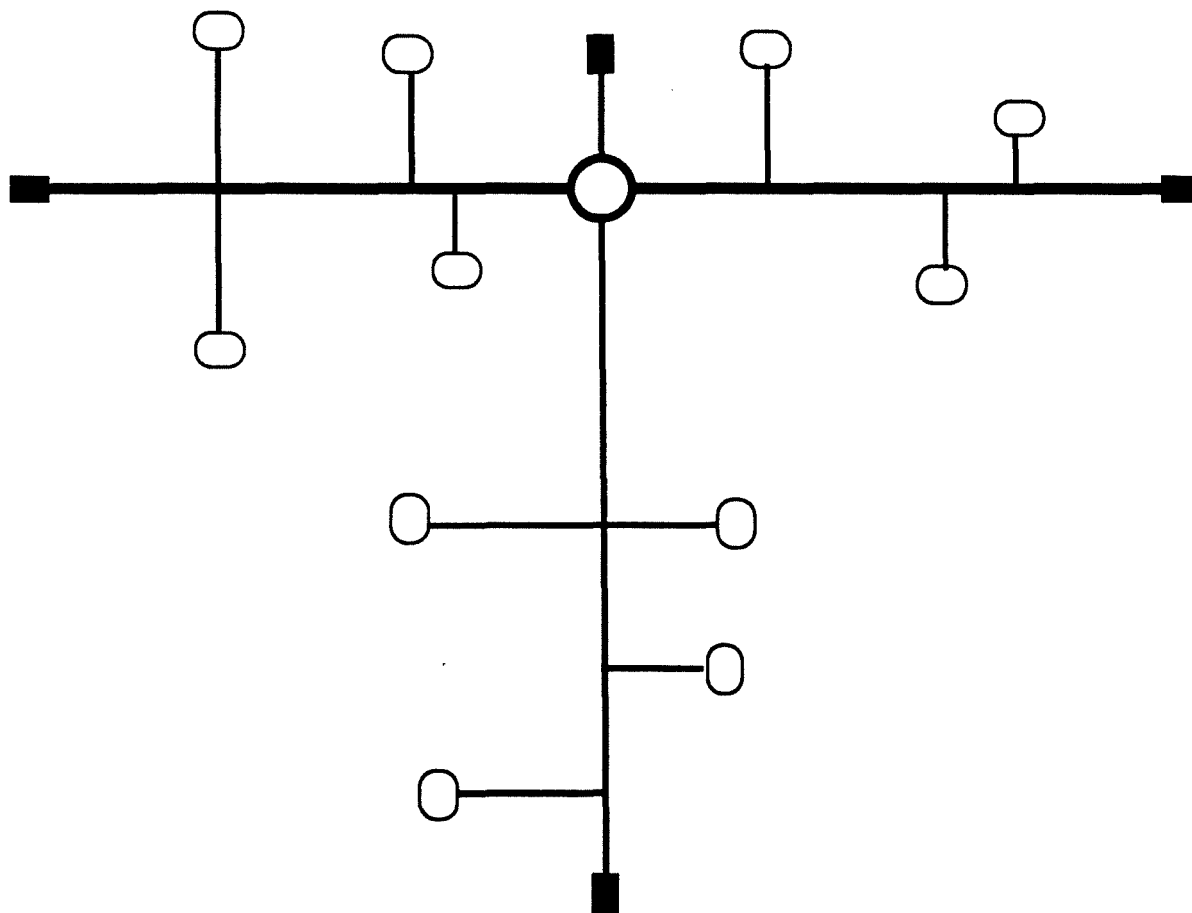


FIG 4 : CONNEXION DE 2 SEGMENTS

La figure 5 représente les deux backbones des Facultés, ceux-ci sont reliés par un câble en fibre optique (figure 5), ce qui nécessite aux deux extrémités de la fibre un mécanisme de conversion du signal optique en signal électrique et vice-versa. Cette conversion est assurée d'un côté par un **remote repeater** et de l'autre par un système plus complexe appelé **bridge**. Cet appareil permet en plus une sélection des signaux à répéter : seuls les signaux destinés à une machine connectée sur l'autre backbone sont transmis, évitant ainsi une surcharge inutile des câbles. Le bridge et le remote repeater sont connectés à leur backbone respectif via un câble spécial au moyen d'un **transceiver**. Le transceiver est une prise de connexion indispensable à tout branchement au câble.

Le Centre de Calcul est relié au même backbone que l'Institut d'Informatique. Les quatre ordinateurs et les deux serveurs de terminaux sont reliés au backbone au moyen d'un **DELNI**, appareil comparable à un gros domino.

L'Institut d'Informatique est connecté au backbone à travers un bridge. Un appareil appelé **DEMPR**, permet l'interconnexion entre le câble venant du bridge et plusieurs câbles plus fins, chacun parcourant un étage de l'Institut d'Informatique. Les machines décrites au point précédent sont connectées à ces câbles au moyen d'un appareil similaire au transceiver appelé **DESTA**.

CHAPITRE 2

LA SECURITE

2.1. APPROCHE CONCEPTUELLE ET OBJECTIFS

2.1.1. INTRODUCTION

La sécurité informatique est un domaine qui recueille l'intérêt d'une vaste couche de la population concernée : informaticiens, utilisateurs, juristes, économistes, Cette soudaine vague d'intérêt répond, dans certains cas, plus à une mode qu'à un besoin réel. Les besoins en matière de sécurité informatique sont fonctions de différents aspects que nous allons aborder.

Ce n'est pas un secret que les ordinateurs ont changé notre façon de travailler. Actuellement, les ordinateurs sont utilisés depuis le bureau de la secrétaire jusqu'au salon du directeur général. Il en résulte que toutes sortes d'informations sont maintenant stockées dans ces ordinateurs, informations qui étaient jadis généralement enfermées sous clé dans une armoire. Les exemples classiques sont les informations personnelles, les prévisions de marché dans une entreprise commerciale ou encore les descriptifs de nouveaux produits. En plus de ces informations à caractère sensible présentes à l'état statique dans l'ordinateur, sont venus s'ajouter les flux des informations transférées via les réseaux de communication. Ces dernières informations ont un caractère nécessitant de plus en plus de confidentialité et d'intégrité, c'est le cas évident des données à caractère financier.

Les possibilités de communication entre machines et de partage des ressources que permettent les réseaux, imposent une utilisation multiusager des systèmes. Ce partage du temps et des ressources de la machine ne fait qu'accroître la problématique de la préservation des données. Le problème premier du responsable de la sécurité est donc de trouver un équilibre judicieux entre d'une part un système totalement sécurisé et d'autre part un environnement offrant une capacité de partage et de communication des informations maximale.

Ce chapitre a pour ambition d'intéresser un vaste public concerné par la sécurité informatique : les informaticiens, les gestionnaires et surtout les utilisateurs, car c'est seulement grâce à leur impulsion et à leurs connaissances de l'environnement de travail que l'on pourra bâtir un plan de sécurité efficace, cohérent et réaliste.

Nous voulons convaincre tous les lecteurs de ne plus prendre pour alibi le caractère multiforme et complexe du risque informatique pour éluder leurs responsabilités en matière de sécurité.

2.1.2. DEFINITION DES CONCEPTS

On regroupe généralement sous le terme de sécurité informatique des concepts très différents. Nous allons essayer de clarifier la situation en définissant les différentes approches existantes. Les méthodes pour aborder le domaine de la sécurité informatique sont multiples, nous avons choisi une approche triple pour définir sa zone d'action : les niveaux de sécurité, les types de risques et les objectifs de la sécurité.

2.1.2.1. LES NIVEAUX DE SECURITE

On peut distinguer trois grands niveaux de sécurité [RAM, 38], la sécurité physique, la sécurité logique et la sécurité sociologique, cette typologie pouvant être au besoin affinée. Ces niveaux pourraient se définir à partir des risques qu'ils doivent contrer.

la sécurité physique assure la protection du matériel, pris au sens large, contre les attaques physiques. Celles-ci sont généralement répertoriées en quatre grandes familles. Les catastrophes naturelles (tremblement de terre, tempêtes, orages, ...), les accidents dûs à l'environnement (incendies, inondations, explosions, ...), les perturbations sociales (guerres, attentats, grèves, ...) et les insuffisances de l'environnement (panne d'électricité, défaillance de la climatisation, etc).

La sécurité logique est le véritable domaine de l'informaticien, elle assure la protection des ressources du système contre les attaques d'origine informatique.

La sécurité sociologique regroupe les moyens de protection des ressources informatiques à caractères humains. C'est le domaine des lois, règlements, us et coutumes,

2.1.2.2. LES TYPES DE RISQUES

Il existe plusieurs typologies des risques, chacune basée sur une approche différente du problème. Nous avons abordé implicitement une première typologie en définissant les niveaux de sécurité. Une autre approche des risques, basée sur le caractère volontaire ou non de leur réalisation, ouvre une seconde dimension au concept de sécurité informatique.

Trois types de risques peuvent alors être considérés [ASS, 1] : les causes accidentelles, les erreurs et les délits.

Les risques accidentels et matériels sont ceux qui entraînent une destruction partielle ou totale de l'équipement, des supports informatiques ou de leur environnement . Il peut également s'agir de défauts temporaires du matériel ou des logiciels.

Les erreurs peuvent se produire à différents niveaux. Il peut s'agir d'erreurs lors de la conception même, lors de l'exécution ou encore de l'utilisation du matériel ou du logiciel, ceux-ci ne présentant à l'origine aucun défaut de conception. Parfois ce sont des erreurs d'encodage, de manipulation ou des erreurs qui se produisent lors de la transmission de données ou lors de l'interprétation ou de l'utilisation des informations.

Les délits ou actes volontaires regroupent les attaques contre l'équipement (vols de matériel ou de logiciel, destruction, ...) et les atteintes au bon fonctionnement du système (introduction de virus, vols d'informations, insertion de fausses données, ...).

2.1.2.3. LES OBJECTIFS DE LA SECURITE

Cette vision curative du problème permet de donner une troisième dimension au concept de sécurité. Les objectifs de la sécurité représentent la réponse aux besoins ressentis par les utilisateurs du système. Trois objectifs principaux constituent la dernière composante de cette vision tridimensionnelle du problème.

La disponibilité vise à garantir à l'utilisateur la présence et le bon fonctionnement de toutes les ressources du système.

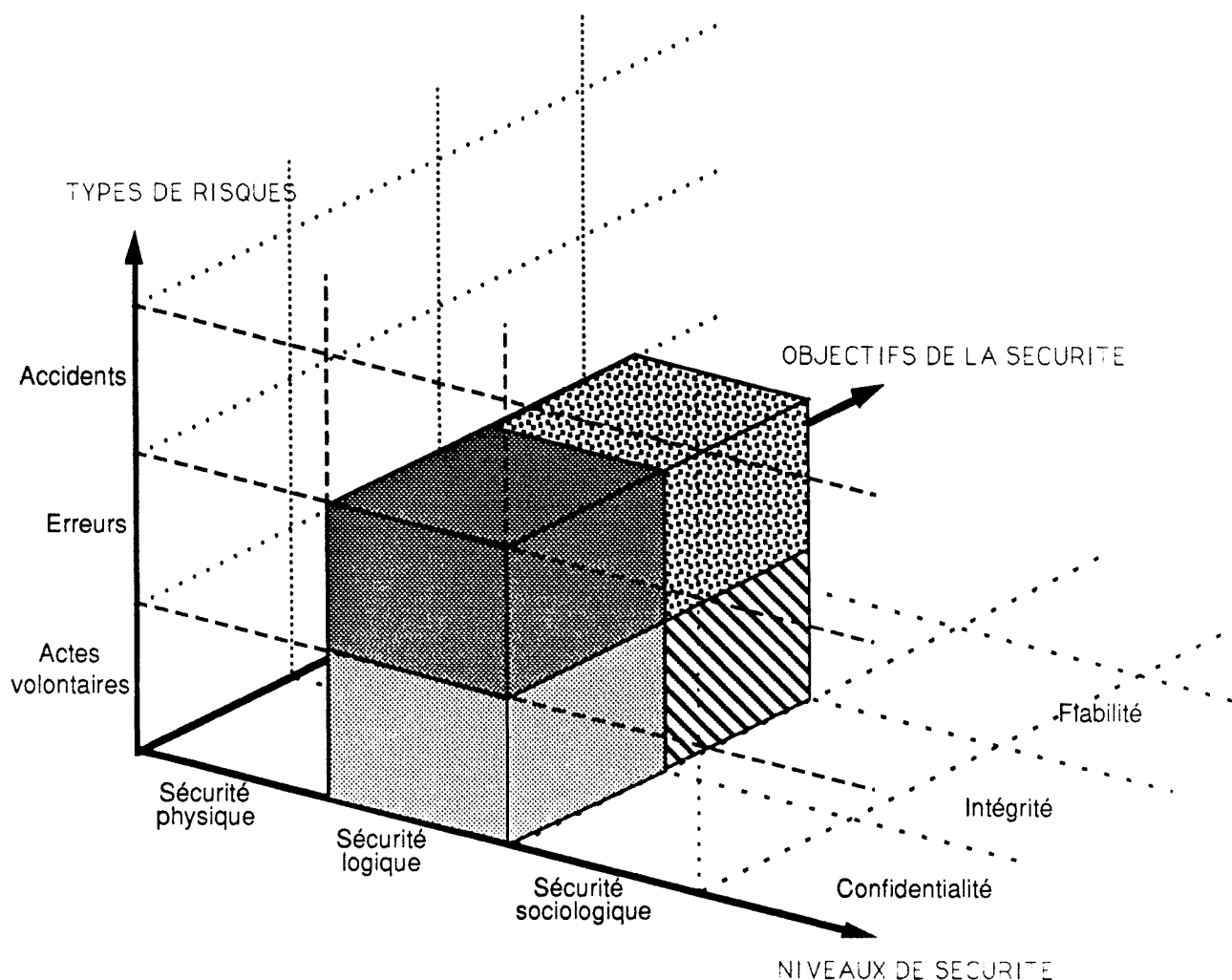
La confidentialité est l'assurance que la capacité de lire les informations est limitée à un ensemble de personnes connues de l'auteur.

L'intégrité est la garantie que la capacité de créer ou de modifier de manière indétectable des données écrites sur un support sécurisé est limitée à un ensemble de personnes connues de l'auteur. La notion de confidentialité et d'intégrité revêt une importance d'autant plus grande que les réseaux de communication véhiculent des informations nécessitant ces propriétés.

2.1.3. OBJECTIFS

Il est évident que nous ne pouvons aborder la problématique de la sécurité informatique dans son ensemble. La nécessité de délimiter une zone d'intérêt particulière, dans l'ensemble des concepts abordés au point précédent, nous semble évidente.

S'il est aisé de se convaincre de cette limitation, il l'est moins de déterminer quels seront les aspects qui seront retenus. Nous avons schématisé les concepts définis dans le point précédent sous forme de graphique tridimensionnel, les différents axes représentant les trois approches évoquées.



Le volume s'élevant au sein du graphique illustre les aspects de la sécurité que nous comptons aborder au sein de ce chapitre. Seul le niveau de sécurité logique nous est apparu intéressant à développer. Ce choix s'est opéré sur base d'une sélection par élimination. Le niveau sociologique, s'il constitue un élément essentiel dans la construction d'une politique de sécurité, reste très éloigné de nos préoccupations d'informaticien. Cette approche sociologique doit de plus être considérée comme minoritaire dans le contexte dans lequel se développe ce mémoire, à savoir le milieu universitaire. Le niveau physique, malgré son importance primordiale même en site universitaire, n'offre que peu d'intérêt dans le cadre de notre recherche. Dans l'approche selon les types de risques, nous développerons les aspects liés aux actes volontaires et aux erreurs, les accidents seront passés sous silence en raison de leur faible causalité en ce qui concerne la sécurité logique. Enfin, du point de vue des

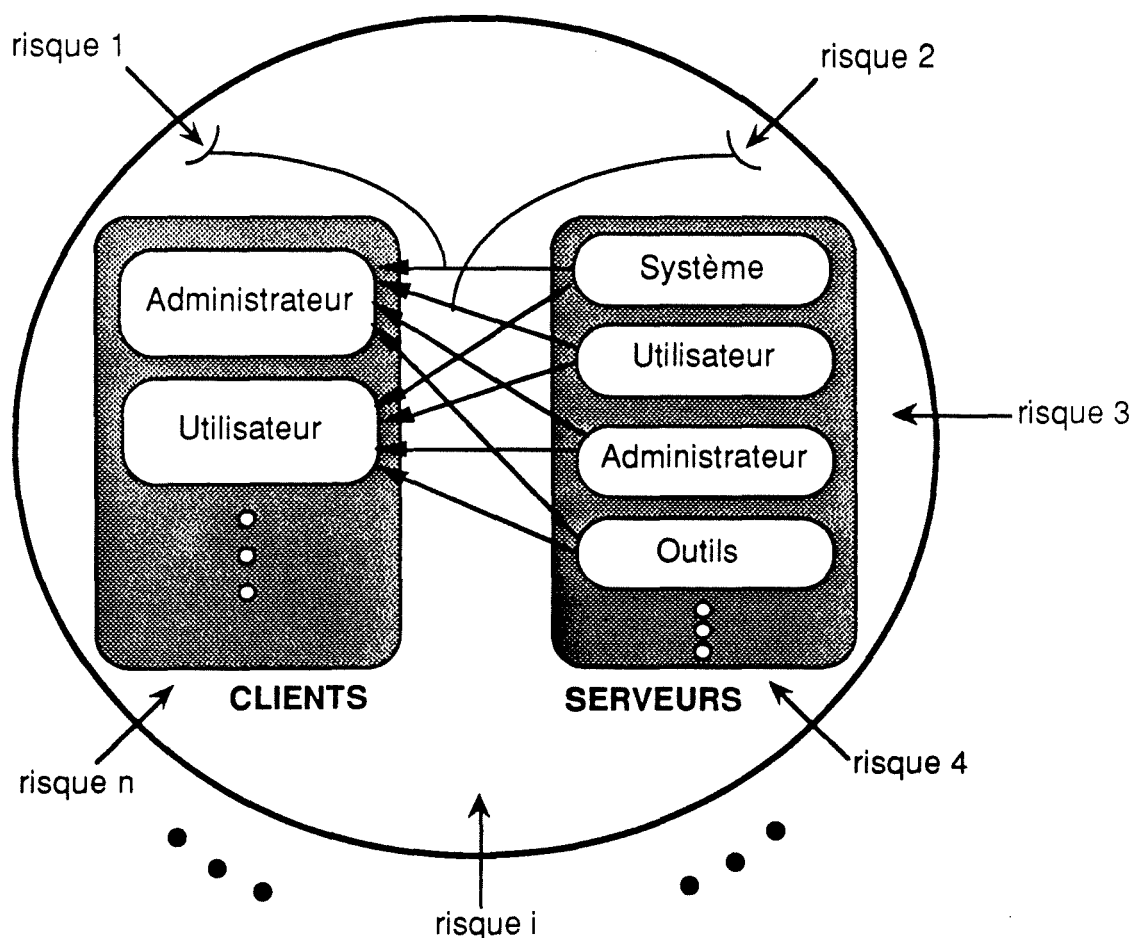
objectifs de la sécurité, nous nous limiterons à aborder les problèmes de protection des informations, sans faire mention de l'aspect fiabilité du système.

Ce choix, même s'il est motivé, n'en reste pas moins arbitraire. Il ne signifie nullement que les aspects de cette problématique passés sous silence ne sont pas digne d'intérêt dans un mémoire, mais seulement que notre attention s'est concentrée sur d'autres.

2.2. MODELE ET METHODOLOGIE

Le modèle sur lequel se base l'analyse de la problématique de la sécurité dans la configuration définie au chapitre 1, est inspiré du modèle client/serveur. Cette modélisation a l'avantage d'être relativement facile à appréhender pour une majorité de lecteurs.

Le schéma qui suit illustre assez clairement cette vision de la sécurité.



Les clients sont les personnes qui expriment un besoin en matière de sécurité. Parmi ces clients, nous allons en distinguer deux principaux : l'administrateur et l'utilisateur.

Les serveurs sont des personnes ou des moyens logiciels ou matériels qui répondent aux besoins exprimés par les clients, en leur fournissant un service de sécurité. Parmi ces serveurs nous distinguerons l'administrateur, l'utilisateur, le système d'exploitation et les logiciels de sécurité.

Ces deux types d'acteurs sont soumis à une série de risques. En offrant un service à un utilisateur, le serveur réduit un ou plusieurs risques de façon totale ou partielle.

Ce modèle va conditionner la méthodologie qui sera suivie tout au long de ce chapitre. Cette méthode peut se décomposer en trois étapes principales : l'analyse des risques et des besoins exprimés par les clients, l'étude des services fournis par les serveurs et les perspectives de développement de services et/ou de serveurs.

2.3. ANALYSE DES RISQUES ET DES BESOINS

2.3.1. LES FACTEURS DE RISQUE

Certains facteurs [RAM,38] augmentent la probabilité de réalisation des risques d'une part, et le nombre de risques d'autre part. Ces facteurs conditionnent donc considérablement leur analyse et leur évaluation.

La concentration des données et de l'appareillage a pour conséquence de faciliter les atteintes à la confidentialité, à l'intégrité et à la fiabilité du système. La compacité des données a la même influence, tout en facilitant la destruction rapide des informations. La faillibilité des programmes est, elle aussi, à l'origine de certains risques, les programmeurs ne garantissant pas l'absence de panne de leur produit. Les réseaux de transmissions de données et la possibilité d'accès à distance aux ordinateurs donnent une dimension nouvelle aux communications et sont à la source de vulnérabilité supplémentaire des informations. La disparition du contrôle visuel, tel qu'il existait dans les organisations non informatisées, est à l'origine d'un accroissement des risques, la nécessité de rediriger cette fonction nous semble d'ailleurs évidente. Tout comme la faillibilité des programmes, la présence de nombreuses copies illicites de logiciels est à la base de problème de sécurité.

2.3.2. TYPOLOGIE DES RISQUES

Une première classification des risques a été présentée lors de la définition des concepts : les fraudes ou actes volontaires, les erreurs et les accidents, ces derniers n'étant pas abordés dans ce mémoire. Plusieurs subdivisions vont nous permettre de raffiner cette première classification.

2.3.2.1. LES ERREURS

Certains auteurs se sont déjà penchés sur la classification possible des erreurs [LAM,30]. On peut distinguer trois types d'erreurs : les erreurs de saisie, les erreurs d'exploitation et les erreurs de conception.

2.3.2.1.1. Les erreurs de saisie

On peut regrouper dans ce type d'erreurs, les erreurs de saisie des opérateurs inhérentes à la nature humaine, les erreurs de transmission et les erreurs d'utilisation des informations.

Les erreurs de saisie ont des conséquences rarement importantes mais constituent un flux permanent de pertes. Sont reprises dans ces erreurs de saisie toutes les erreurs de lecture, d'interprétation ou d'application des règles. Ce type d'erreurs nous paraît toutefois à la limite du domaine informatique pour une raison évidente. Il existait bien avant l'utilisation de l'outil informatique et l'usage de l'ordinateur en a même diminué le nombre.

Les erreurs de transmission dépendent de la technique employée. L'utilisation massive des réseaux d'ordinateurs risque d'augmenter la part de ce type d'erreurs. Elles nous intéressent particulièrement étant donné la configuration qui fait l'objet de notre étude.

Les erreurs d'utilisation de l'informatique sont plus subtiles. Elles sont basées sur la déviation de l'interprétation des informations issues d'un traitement informatique, par rapport à celles issues d'un traitement purement humain.

Bien que les pertes soient difficiles à isoler et à évaluer, il convient de ne pas sous-estimer cette catégorie à laquelle on a la faiblesse de trop s'accoutumer [LAM,30].

2.3.2.1.2. Les erreurs d'exploitation

Ces erreurs peuvent recouvrir des formes très variées telles que oubli ou effacement d'un fichier, la mauvaise manipulation d'un matériel ou l'utilisation erronée d'une commande. Il est parfois malaisé de classer ce type d'erreurs que peuvent occasionner une négligence, un excès de laxisme, une faute professionnelle ou encore un acte volontaire de malveillance.

2.3.2.1.3. Les erreurs de conception

Une erreur de conception peut avoir des conséquences dramatiques sur la conformité des résultats ou sur les performances des traitements qui les fournissent. Ce type d'erreurs reste toutefois en dehors de notre zone d'intérêt car ses conséquences touchent généralement la fiabilité du système plutôt que la protection des données.

2.3.2.2. LES FRAUDES ET ACTES VOLONTAIRES

Les actes volontaires et les fraudes peuvent être analysés suivant une double approche, d'un côté l'attention est portée sur la menace que représente un risque et d'un autre côté sur la technique utilisée pour réaliser l'attaque.

2.3.2.2.1. Les menaces

Ces menaces peuvent généralement être classées en deux familles : les menaces passives et les menaces actives. Les menaces passives sont celles dont la mise à exécution n'apporte aucune modification de l'état initial du système, contrairement aux menaces actives qui engendrent des modifications partielles ou totales, temporaires ou permanentes du système [GAS,23].

L'observation est la principale menace passive, elle consiste en l'acquisition de l'information par une personne non-autorisée. Cette simple indiscretion dont les conséquences s'avèrent souvent difficilement chiffrables, sont d'ordre déontologique voir pénal. Son importance ne fait que croître dans le monde commercial et industriel où l'espionnage constitue une arme ultime contre la concurrence.

L'inférence est une menace passive, elle est basée sur la déduction d'informations utiles à partir des caractéristiques de données protégées contre l'observation directe.

La **modification** est une menace active où l'attaquant modifie des informations dont l'accès lui est interdit.

L'**insertion** est une autre menace active qui consiste en l'ajout d'informations par une personne non-autorisée, tout en faisant croire que cette information a été écrite par son auteur légitime.

La **mascarade** est une menace d'attaque active ayant pour but, pour une personne non-autorisée, de s'identifier à une autre personne.

La **répétition** peut être considérée comme une double menace d'observation et d'insertion. Elle permet à son auteur d'insérer une information identique à celle observée précédemment. Cette menace existe même pour une information protégée contre la lecture. Un exemple simple pour lequel la répétition peut être très utile est l'ensemble des transactions financières.

Le **refus de service** constitue une large classe de menaces. Dans le domaine des communications, l'attaque la plus souvent mentionnée est la surcharge artificielle d'un réseau par l'insertion massive d'informations inutiles. Cette menace se rapporte plus souvent au problème de la fiabilité du système qu'à la protection des données, elle sort donc de notre sphère d'intérêts mais n'en reste pas moins très présente.

Le **vol de temps informatique** consiste en l'utilisation sans autorisation de la capacité informatique d'autrui à des fins personnelles [ASS,1].

Ces différentes menaces peuvent parfois être replacées dans un contexte criminogène plus vaste et avoir des conséquences lourdes pour l'organisation qui en est la victime. Nombreuses sont en effet les atteintes à la sécurité informatique ayant pour but premier une escroquerie, une extorsion, un détournement ou encore un faux. Il n'est pas inutile de rappeler que l'environnement dans lequel se réalise ce travail se prête difficilement à ce genre de raisonnement et que nous n'y accorderons donc pas plus d'attention.

2.3.2.2.2. Les techniques d'attaque

Pour pouvoir se protéger efficacement, il est non seulement utile de bien connaître la menace mais aussi la technique qui permet de mettre cette menace à exécution. La liste des techniques que nous allons évoquer est loin d'être exhaustive, sa taille n'a de limite que celle de l'imagination. Très souvent ces méthodes sont d'ailleurs combinées, ce qui complique encore plus leur détection.

L'**intrusion** ("hacking") est le nom communément donné à la technique qui consiste à deviner par essais successifs la clé d'authentification d'un utilisateur légitime d'un système. L'intrusion peut se faire dans un système entier ou dans des domaines du système offrant certaines fonctionnalités auxquelles l'utilisateur n'a pas droit. L'intrus dans ce cas est un utilisateur légitime du système global.

Le **wiretapping** est une technique qui permet de se connecter de façon pirate sur un support de transmission de données. Deux méthodes existent pour accéder à l'information circulant sur le moyen de transmission, une connexion physique directe ou une captation par induction. Si la première des deux méthodes permet aussi bien l'observation que l'insertion d'informations, il est évident que la seconde n'autorise que l'observation. Trois types de support sont généralement utilisés : la paire torsadée, le câble coaxial et la fibre optique. La facilité et l'efficacité de la connexion dépendent d'une part du type de support, et d'autre part de la protection des données qui transitent sur ce support.

Le tableau qui suit illustre les possibilités de connexion propres à chaque support.

SUPPORT	connexion physique	induction
paire torsadée	oui	oui
câble coaxial	oui	non
fibre optique	non	non

Le **virus**, technique très en vogue depuis quelques temps, infecte les programmes ou les données du système en les modifiant de façon arbitraire, de telle sorte que ceux-ci véhiculent à leur tour des virus. Le virus est donc une section de programme capable sous certaines conditions de se copier lui-même selon un mécanisme similaire à la mitose cellulaire.

L'exemple classique est la présence d'une section parasite dans un programme exécutable ou dans une procédure de commande. Lors de l'exécution de cette procédure le virus va voir la liste des fichiers qui lui sont accessibles, il vérifie si ces fichiers sont déjà infectés et, s'ils ne le sont pas, il copie la section parasite dans le fichier en question. Après un certain nombre d'exécutions de différents programmes ou de différentes procédures de commande, pratiquement tous les fichiers contiendront la section parasite. Un fichier de données n'est pas dangereux pour la transmission des virus, sauf dans le contexte d'un système d'exploitation dont on ne maîtrise pas totalement les opérations. D'autre part, certains éditeurs de texte courants exécutent toujours la première ligne du fichier de données.

Le but d'un virus est généralement d'interrompre certains services ou de provoquer des modifications des informations. Il peut en effet être porteur de commandes néfastes telles que l'effacement de tous les fichiers. Le second effet est un accroissement parfois important de l'occupation des mémoires de masse. Les conséquences toucheront donc la fiabilité du système mais également la protection des informations et plus particulièrement leur intégrité.

Le **ver** ("worm") est un processus qui se déplace constamment d'un endroit de la mémoire à un autre ou d'une machine à une autre via les moyens de communication. Le ver est né d'une idée naïve et positive qui était d'utiliser le temps perdu la nuit sur des réseaux de

micro-ordinateurs, au moyen d'un processus qui cherche où il peut récupérer de la puissance. C'est donc un processus qui doit être lancé et qui essayera ensuite d'accéder à un maximum de ressources et de se développer dans toutes les machines accessibles. La conséquence est double, tout d'abord il prend de la place et de la puissance sans aucun profit, ensuite il peut créer des processus néfastes, voir installer un virus. Le virus était confiné à une seule machine, la contamination ne se faisant pas de façon automatique (copie de disquettes, de fichiers, etc), le ver a lui une capacité de propagation autonome. Son utilité est très grande pour le fraudeur qui ignore où se trouvent les informations profitables et les pseudo-informations [ASS,1]. Ce programme a donc pour but de scanner les différentes zones de la mémoire dans le but de trouver les informations pertinentes.

L'introduction de vers est aussi une technique logicielle utilisée par certains fabricants de programme pour prévenir les copies pirates. Ce sont des "bugs" intentionnels qui sont automatiquement introduits sur toute copie illégale du programme et qui auront une conséquence corruptrice sur son exécution ou sur les données qu'il manipule [POU,35].

L'"**entrée par l'arrière**" est un accès qui n'est généralement connu que du concepteur mais qui est parfois découvert par d'autres [ASS, 1]. Ce terme référence la technique qui permet d'allouer un contrôle externe sur un programme au travers de l'inclusion d'une section de code. Certaines "portes" sont codées en toute légitimité pour aider au développement et à la maintenance du programme, cette technique a aussi été utilisée pour obtenir un accès illégitime à un équipement ou à des données. Ces portes sont quelques fois utilisées par les ingénieurs systèmes pour accéder au système à distance, et leur permettre d'examiner la machine en fonctionnement dans un but de maintenance préventive. La sécurité du système est donc tributaire de celle de l'ingénieur, de telle sorte que des contrôles adéquats de cette facilité doivent être assurés avec l'accord de la firme qui assure ce service de maintenance [POU,35].

La **bombe logique**, ou encore appelée bombe à retardement, est un programme dont l'exécution est déclenchée par la réalisation d'un événement particulier (une certaine heure, une certaine date, la présence d'un utilisateur particulier dans le système, l'état d'un fichier, ...). Son principe est similaire à celui du virus mais ne comprend aucune reproduction. Les conséquences du point de vue de la sécurité peuvent être tout aussi désastreuses.

Le **cheval de Troie** est un programme qui simule l'exécution d'une commande ou d'un processus connu de la victime, tout en y ajoutant l'exécution d'une ou plusieurs autres commandes qui mettent en péril la sécurité de celui qui l'a activé ou d'une tierce personne. Les conséquences sont similaires à celles d'une bombe logique et laissent libre cours à l'imagination de son auteur.

Une classification totalement indépendante peut être basée sur le type de perte qu'engendrerait la réalisation des risques. Quatre types de pertes peuvent être envisagés [LAM,30].

Les dommages matériels et annexes reprennent les coûts de réparation et de remplacement des matériels et logiciels endommagés, à ceci peuvent s'ajouter une série de frais indirects (expertise, reconstruction, ...).

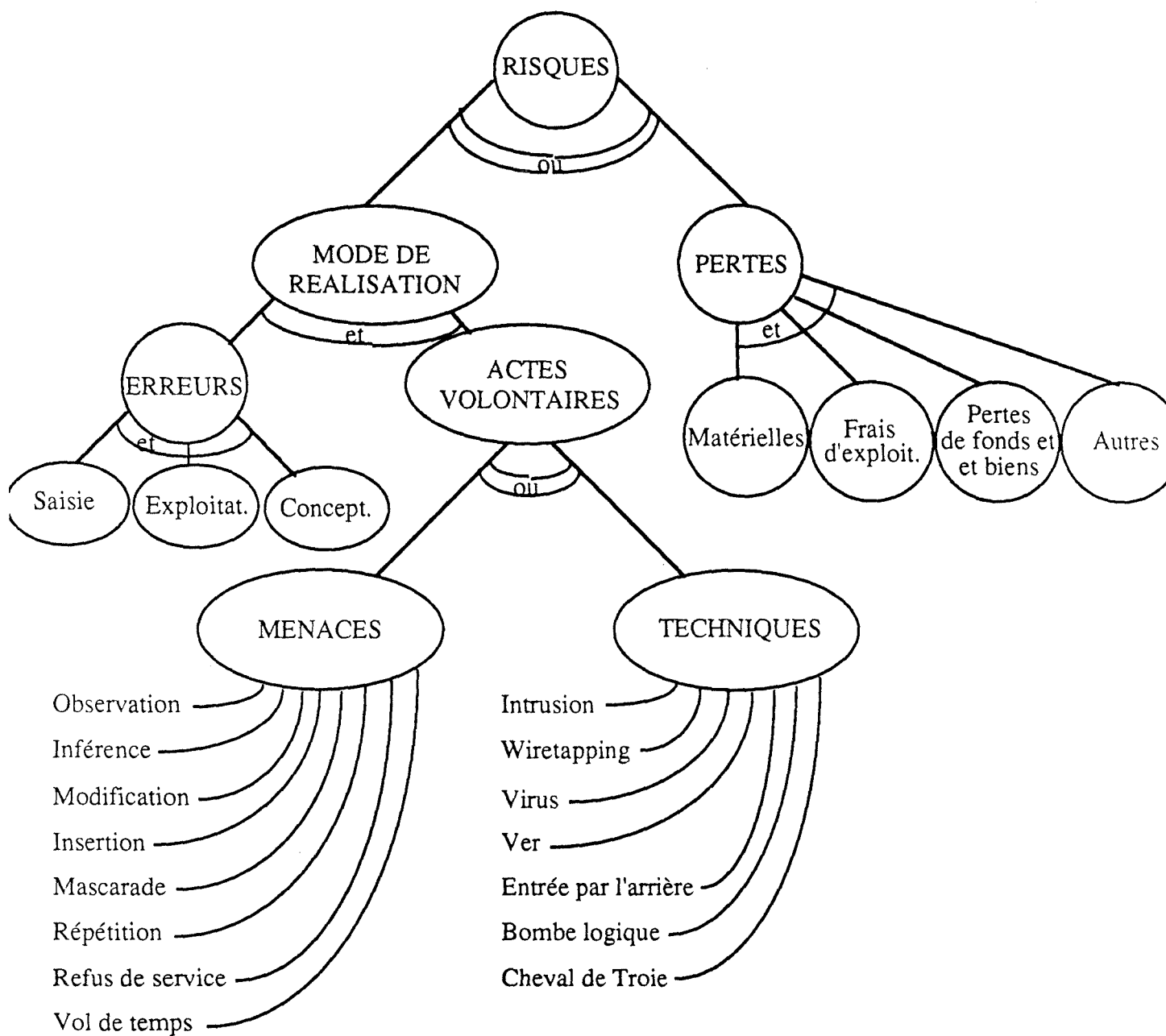
Les frais supplémentaires correspondent au surcoût engendré par la mise en oeuvre de moyens de secours informatiques et non-informatiques (location, heures supplémentaires, ...). Les pertes d'exploitation sont dues à la baisse de la marge bénéficiaire brute de l'entreprise induite par le sinistre.

Les pertes de fonds et pertes de bien correspondent à la disparition des liquidités et des biens propres (stockés ou non).

Les autres pertes constituent une catégorie de pertes dont le contenu est très varié et qui reprend les préjudices corporels, les frais de justice,

Il est évident que cette classification s'adapte très bien au milieu industriel et commercial, mais n'est pas du tout rentable dans un contexte universitaire.

Les différentes classifications proposées peuvent être représentées par un arbre "et/ou" dans lequel un noeud correspond à une typologie, un arcs "et" à une subdivision et un arc "ou" à un choix de typologie.



2.3.3. EVALUATION DES RISQUES

Cette étape correspond, dans la méthodologie proposée dans le cadre du cours de sécurité [RAM,38], au relevé des risques potentiels auxquels est soumis le cas traité.

Une véritable quantification des risques nous paraît hors de portée eu égard au contexte de ce mémoire. Les pertes qui résulteraient de la concrétisation de ces risques sont impossibles à chiffrer, contrairement au cas d'une entreprise à caractère industriel ou commercial. Nous nous contenterons donc de les aborder en détails et d'évaluer de façon qualitative et non quantitative les conséquences qui pourraient en résulter.

Nous allons développer parmi les risques énumérés dans les typologies, ceux applicables au site universitaire en général et à la configuration telle qu'elle a été décrite en particulier. Feront donc l'objet d'un développement tout au long de cette partie, les risques d'erreurs d'exploitation et les risques basés sur des actes volontaires malveillants.

2.3.3.1. LES ERREURS

Seules, à notre sens, les erreurs d'exploitation présentent un risque potentiel pour la configuration étudiée. Les risques d'erreurs de saisie tels qu'ils ont été définis nous paraissent minimes en raison de l'absence quasi-totale de fonctions de saisie de données, au sens strict du terme, dans le système. Les seules opérations de saisie ponctuelles qui sont censées se réaliser s'apparentent plutôt à des opérations d'exploitation. Les conséquences qui découlent de telles erreurs peuvent être comprises dans celles découlant des erreurs d'exploitation. Un exemple simple de ce type particulier d'erreur de saisie pourrait être la saisie du nom d'un fichier comme argument dans l'activation de la commande d'effacement *rm* en Unix. Ce type d'erreur peut avoir une conséquence non négligeable sur la sécurité et plus particulièrement sur l'intégrité des données.

Sous le terme d'erreur nous regrouperons toutes les failles dues aux négligences, insouciances et inconsciences de l'utilisateur ainsi que toutes les erreurs de manipulation, de frappe et de méconnaissances des règles de fonctionnement du système. Certaines erreurs n'ont pas de conséquences actives directes sur la sécurité, mais représentent un

catalyseur d'attaques, ces erreurs sont, dans pratiquement tous les cas, commises par la victime elle-même. Les autres erreurs, étant celles dont les conséquences sur la sécurité sont directes, sont tout autant commises par la victime que par d'autres personnes. L'évaluation des risques va se baser sur cette division entre ces deux types d'erreurs que nous nommerons respectivement erreurs indirectes et erreurs directes.

2.3.3.1.1. Les erreurs indirectes

(a) Gestion du mot de passe

Le choix du mot de passe est d'une importance capitale; en effet, c'est lui qui constitue le premier et souvent le seul rempart contre les malveillances. Utiliser un mot de passe évident à déduire, le stocker dans un endroit accessible, omettre de le modifier ou le taper à la vue d'une autre personne constitue autant de facteurs accroissant le risque d'attaque. Nous verrons ultérieurement les règles à suivre pour diminuer l'importance de ce risque.

(b) Accès aux données

Les permissions d'accès à un fichier déterminent qui peut y accéder et ce qu'il peut faire. Le fonctionnement de ces permissions sera expliqué en détail dans le point traitant des services de sécurité rendus par les différents serveurs.

Les répertoires disposent également d'un mécanisme de contrôle d'accès travaillant de façon similaire à celui des fichiers. Le fonctionnement de ce mécanisme est généralement mal connu des utilisateurs, bien que son ignorance puisse faire courir d'importants dangers. En effet, une mauvaise protection d'un répertoire peut rendre possible sa modification, n'importe qui peut alors remplacer (*mv*) ou effacer (*rm*) un fichier de ce répertoire, et ce, qu'elle que soit la permission d'accès au fichier lui-même.

Les permissions par défaut lors de la création d'un fichier peuvent être fixées par l'utilisateur. Leur mauvaise utilisation rend vulnérable le contenu de chaque nouveau fichier ou répertoire.

L'accès à certaines données peut n'être possible que par le biais de l'exécution d'un programme dit programme SUID, cet accès est donc tributaire des permissions accordées au programme en question. Il est donc nécessaire d'être conscient des risques que comporte une telle technique d'accès aux données.

Il est possible en Unix de créer un lien vers un fichier existant grâce à la commande *ln*, ce lien permettant de référer au fichier original par un nouveau nom. Le problème au niveau de la sécurité intervient à nouveau dans le cas d'un programme SUID, en effet si des liens ont été créés et qu'on efface le fichier, il restera accessible à partir des liens établis.

(c) Transfert de données

La commande *UUCP* qui permet le transfert de fichiers entre deux systèmes UNIX, peut avoir des conséquences néfastes au niveau de la sécurité si on ne connaît pas son fonctionnement exact. Lors d'un transfert de fichier, ce dernier est placé dans un répertoire standard. Ce répertoire offre toutes les permissions d'accès pour tous et les fichiers qui y sont placés appartiennent à UUCP et sont accessibles en lecture et écriture par tout utilisateur. Le simple fait de protéger ces fichiers en lecture et écriture pour toute autre utilisateur que soi-même ne suffit donc pas pour contrer toute attaque.

Plusieurs autres systèmes de transfert de fichiers utilisent une technique similaire se basant sur l'utilisation d'un répertoire qui doit être accessible à tous et donc envers lequel il est nécessaire de prendre des précautions identiques.

(d) Les fichiers temporaires

Le système Unix utilise deux répertoires particuliers pour y placer les fichiers temporaires issus de processus tels que le tri ou la compilation C. Ces répertoires étant accessibles en recherche, lecture et écriture par tout utilisateur, certains problèmes de confidentialité et d'intégrité peuvent se poser.

Tout d'abord, le fait que la commande *sort* sauve une version provisoire d'un fichier à trier dans ce répertoire, peut avoir des conséquences fâcheuses si ce fichier contient des informations sensibles.

Un deuxième problème peut se poser pour un programmeur qui utilise ces répertoires pour y placer ses propres fichiers temporaires.

(e) La messagerie

Certains terminaux intelligents interprètent des séquences de caractères bien particulières comme si elles avaient été introduites au terminal par l'utilisateur. C'est notamment le cas du "!" interprété par le *mail* comme le début d'une commande shell, le shell est donc déclenché et il interprète le reste de la ligne comme une commande. Il est aisé d'imaginer les conséquences désastreuses que peut avoir un message du type *! rm -r ** .

(f) Les changements

Certains changements apportés par l'administrateur du système peuvent s'avérer être à l'origine de l'accroissement d'un risque existant ou de l'apparition d'un nouveau risque. Ces changements qu'ils soient orientés logiciel (changement de version du système d'exploitation) ou orientés matériel (ajout d'une station ou d'un serveur) et aussi minimes soient-ils, doivent être pris avec le plus grand sérieux . Ils sont généralement la cause d'un impact non négligeable sur la sécurité et le contrôle interne . Ces changements peuvent parfois être tout à fait insignifiants, comme l'ajout d'un utilisateur au système ou la modification des quotas disque

d'un autre utilisateur, et pourtant avoir une influence sur la problématique de la protection des données. De tels changements nécessitent une analyse en détail des répercussions qu'ils peuvent avoir.

(g) Les connexions

Lorsqu'un utilisateur établit une connexion à un hôte, il le fait généralement soit par une station de travail reliée au réseau Ethernet, soit par un terminal connecté au serveur de terminaux lui-même relié au réseau. La pratique classique, et d'ailleurs recommandée par le Centre de Calcul, pour la salle des terminaux, est de couper la connexion en activant un "break", ce qui supprime le circuit ouvert depuis le terminal jusqu'à l'hôte. Un problème de sécurité se pose toutefois dans deux cas de figure que nous avons eu l'occasion de vérifier, l'existence de ces deux cas répertoriés ne limitant nullement la possibilité d'existence d'autres tout aussi graves.

Le premier cas a trait à l'utilisation du remote-login à partir d'un terminal. Cette technique permet d'entrer dans plusieurs systèmes en cascade et donc de créer plusieurs connexions successives et dépendantes. Si au moment de terminer la session sur le *n*ème hôte l'utilisateur actionne la touche "break", il coupe en réalité la ligne vers le serveur de terminaux et éventuellement celle vers le cluster Quick si ce dernier était son hôte de départ. Mais il laisse accessible toute la ligne reliant les hôtes ayant fait l'objet d'un remote-login. Cette connexion, toujours existante, peut être reprise par un autre utilisateur qui obtiendra ainsi un accès à tous les systèmes reliés par cette connexion, il est aisé d'évaluer le risque que représente une telle erreur et les conséquences qu'elle peut avoir.

Le second cas de figure se présente en cas d'utilisation du logiciel de transfert de fichier **KERMIT**. Après le lancement de l'application sur l'hôte éloigné, l'utilisateur lui assigne un rôle au moyen d'une commande telle que *server* qui signifie que cet hôte est considéré comme un distributeur de fichier. Un caractère de contrôle ou une commande de menu sur

Macintosh doit être activé pour mettre fin à ce rôle. Si l'utilisateur oublie d'actionner cette fonction ou que celle-ci ne répond plus et l'oblige à relancer son micro-ordinateur, la connexion ouverte restera présente malgré la sortie du logiciel et même malgré le relancement de la machine. Les conséquences d'une telle situation sont donc identiques à celles mises en exergue dans le cas de figure précédent.

2.3.3.1.1. Les erreurs directes

Les erreurs directes sont donc celles ayant un impact actif et immédiat sur la sécurité et pouvant être commises par la victime elle-même ou par une tierce personne. Nous évoquerons cette catégorie à travers la description des erreurs possibles.

(a) L'effacement de données

L'effacement d'un fichier par inadvertance est un cas fréquemment rencontré dans l'environnement Unix. La commande *rm* permet d'effacer irrémédiablement un fichier sans offrir la possibilité à l'utilisateur de confirmer cette opération ou de récupérer son bien par une autre commande. On peut mettre en exergue pour ce problème deux cas de figure assez typiques.

Lorsque l'utilisateur possède plusieurs fichiers ayant des noms relativement proches (parfois une seule lettre change), il court le risque de faire une regrettable faute de frappe et se voir privé de données importantes.

Le second cas de figure illustre le problème de la proximité des noms, pouvant également occasionner une faute d'interprétation, en ce sens que l'utilisateur lui-même ne sait plus vraiment à quoi correspond chaque fichier. Cette apparentement des noms n'est pas nécessaire pour que ce risque soit présent, il peut s'agir d'un manque de connaissances de ses données.

Nous avons vu que la victime potentielle d'une telle erreur pouvait aussi en être l'auteur, mais une tierce personne peut

également en être l'instigatrice. Les mêmes cas de figures peuvent être évoqués, mais ce raisonnement suppose que cette tierce personne ait un accès au répertoire de la victime. Cette solution est logiquement envisageable si par exemple cette personne fait partie d'un groupe particulier d'utilisateur ayant des permissions d'accès plus importantes.

Quelques commandes Unix peuvent aussi avoir pour conséquence indirecte l'effacement d'un fichier.

C'est le cas de la commande de copie *cp* qui provoque le remplacement du fichier destination, si celui-ci existe, par le fichier origine, entraînant la perte éventuelle de données importantes.

La commande de modification du nom d'un fichier *mv* comporte les mêmes risques, si le fichier de destination existe. De plus, si le changement de nom s'accompagne d'une mutation du fichier vers un autre système de fichier, l'opération correspond à une copie avec effacement de l'original.

Le problème d'effacement de fichier est aussi présent lors de l'utilisation de la commande *cpio*. Cette commande permet d'archiver la structure d'une hiérarchie de répertoire dans un fichier ordinaire. Elle permet inversement d'extraire d'un fichier archive la structure d'une hiérarchie. Si l'option *-u* est utilisée, tout fichier existant sera écrasé par le fichier de même nom extrait du fichier archive, s'il est plus récent que ce dernier. Il est inutile de préciser les dégâts que pourrait occasionner, dans certains cas, la simple utilisation de la commande *cpio -u*

2.3.3.2. LES ACTES VOLONTAIRES

Les actes volontaires recouvrent deux types de fraudes informatiques : l'attaque passive et l'attaque active. Cette distinction a déjà fait l'objet d'une explication lorsque nous avons abordé les types de menaces.

2.3.3.2.1. Les attaques passives

Comme nous l'avons déjà vu, l'attaque passive est généralement réalisée par observation ou par inférence. Parmi c'est deux actes d'attaques, c'est l'observation qui retiendra le plus notre attention, l'inférence exigeant de son auteur des moyens sans aucune mesure avec les gains qu'il pourrait en retirer. En effet, les réseaux informatiques universitaires recèlent rarement des données de très grande valeur pour un attaquant extérieur.

(a) L'accès aux fichiers

Les fichiers non protégés constituent une cible de choix pour l'attaquant, il pourra consulter les informations qu'ils contiennent s'ils ne sont pas protégés en lecture.

L'absence de protection des répertoires offre au fraudeur la possibilité de réaliser toute une série d'attaques, même si le fichier visé est totalement protégé. La plus évidente étant la simple extraction d'un fichier par copie.

La copie d'un fichier ou d'un répertoire a aussi pour conséquence de copier toutes les permissions du fichier original, du moins si le fichier de destination n'existe pas. Seul le nom du propriétaire est modifié mais le SUID, s'il existe, n'est pas modifié; ce qui a pour effet de l'associer à un autre utilisateur et de lui octroyer les accès qu'avait l'utilisateur original.

Comme vu au point précédent, la copie de fichiers avec liens, la copie de fichiers situés dans le répertoire utilisé pour le transfert des données et des fichiers temporaires, sont autant de possibilités d'indiscrétion.

Une autre atteinte possible à la confidentialité des informations peut être réalisée au moyen de la commande d'archivage et désarchivage *cpio*. Lors de son utilisation, le fichier archive contient non seulement toutes les informations

des répertoires mais aussi les informations de chaque fichier, à savoir propriétaire, groupe, dernière date de modification, etc. Ce qui a pour effet que les fichiers créés à partir d'un fichier archive reçoivent les permissions d'accès stockées dans le fichier archive. Par contre le propriétaire et le groupe d'un fichier créé à partir d'un fichier archive ne sont pas ceux qui y étaient stockés mais bien ceux de l'utilisateur qui a lancé la commande *cpio* d'extraction, sauf si celui qui la lance est superutilisateur.

(b) L'écoute du réseau

L'écoute du réseau peut s'avérer un moyen efficace pour observer les informations confidentielles, au moyen d'une des techniques évoquées au point précédent. Cette écoute n'est pas à négliger, vu la configuration à caractère distribué que nous connaissons au sein de l'Institut, cette importance étant renforcée par le fait qu'une majorité de stations de travail sont dépourvues de mémoire de masse, ce qui implique un transfert de toutes les données traitées via le câble Ethernet.

Cette possibilité d'attaque demeure néanmoins peu probable en raison d'une part des nombreuses autres facilités moins onéreuses et plus discrètes, et d'autre part du peu de rentabilité d'une telle attaque.

2.3.3.2.2. Les attaques actives

On peut distinguer deux types d'attaques actives, celles à effet direct et unique et celles dont le premier effet a pour but d'en préparer un ou plusieurs autres, le cheval de Troie étant l'exemple classique d'attaque indirecte.

(a) L'attaque active directe

Le cas classique d'attaque directe est la destruction ou la modification d'informations appartenant à autrui en profitant des

erreurs commises par la victime dans la protection de ses fichiers.

Un fichier avec de larges permissions d'accès, notamment en écriture, autorise n'importe quel autre utilisateur à modifier son contenu. Le même risque se présente pour les répertoires. L'attaquant pourra, par exemple, détruire les fichiers ou remplacer un de ceux-ci par un faux. Pour ce faire, il copie sa version en utilisant le même nom que celui du fichier cible. Il ne lui reste plus qu'à modifier le mode d'accès, le propriétaire et éventuellement le groupe du fichier copié pour masquer totalement son méfait.

(b) L'attaque active indirecte

L'attaque active indirecte est composée d'une attaque de base mettant le système dans un état particulier favorisant la réalisation d'un ou plusieurs autres attaques passives ou actives.

L'attaque sur le mot de passe est par définition une attaque indirecte, en ce sens que la découverte du mot de passe d'un utilisateur n'a d'intérêt que si on l'utilise.

Le mot de passe de chaque utilisateur, afin qu'il soit disponible pour le système, est stocké dans un fichier. Ce fichier lisible par tout utilisateur, est bien sûr protégé en écriture et la partie contenant le mot de passe est chiffrée, ce qui rend son utilisation impossible même pour le superutilisateur.

Lors de l'entrée d'un utilisateur, le système compare la version chiffrée du mot de passe introduit avec celle se trouvant dans ce fichier. Le fichier n'étant pas accessible en écriture, il est impossible de modifier la version chiffrée.

Mais un attaquant pourrait générer automatiquement des chaînes de caractères, les chiffrer et les comparer avec le contenu de ce fichier. S'il en trouve une qui correspond, elle lui permettra d'entrer à sa guise dans le système. Cette attaque est évidemment facilitée par des choix de mots de passe, de la part des utilisateurs, trop courts ou faciles à deviner.

Il existe un fichier très sensible au point de vue sécurité en Unix, en ce sens qu'il est exécuté à chaque entrée dans le

système. Il peut être très intéressant pour une personne malveillante de modifier son contenu. La liste d' exemples qui suit est loin d'être exhaustive :

- modifier le masque de permission d'accès par défaut
- modifier le chemin courant des commandes
- modifier le chemin du courrier
- copier tous les fichier

Un des exemples les plus dangereux est la copie du shell à partir de ce fichier. Ceci permet au bénéficiaire de la copie, après avoir modifié les permissions d'accès, d'exécuter n'importe quelle commande dans le shell du propriétaire du .profile, ce qui donne au fraudeur les mêmes privilèges que s'il disposait du mot de passe.

Certains éditeurs dont vi utilisent un fichier d' initialisation dont le fonctionnement est similaire à ce fichier.

Le cheval de Troie constitue sans doute la plus grande menace d'attaque indirecte. Un principe très répandu de fonctionnement du cheval de Troie est de faire chercher la commande classique dans un répertoire autre que le répertoire système en premier lieu. La fausse commande ayant le même nom sera exécutée en lieu et place de la véritable. L'exemple suivant nous montre le cas d'une fausse commande *su* qui permet à celui qui l'a implémentée de recevoir dans son courrier le mot de passe de la victime :

```
stty -echo
echo "password : \c"
read X
echo ""
stty echo
echo $1 $X | mail outside ! dedo &
sleep 1
echo Sorry
rm su
```

Ce cheval de Troie simule donc la commande *su* en respectant parfaitement son protocole de fonctionnement, faisant croire à son utilisateur qu'il a commis une erreur dans la frappe de son mot de passe. On peut remarquer que la détection à posteriori est impossible puisque le faux programme s'efface après son exécution. Il est clair que, pour permettre son exécution, ce programme a dû être placé dans le premier répertoire scanné par le système. La technique utilisée varie entre deux extrêmes. Soit on apporte très peu de modification au chemin du fichier d'initialisation, mais dans ce cas le cheval de Troie sera placé dans un endroit "visible". Soit on place le cheval de Troie dans un endroit "invisible" (le répertoire de l'attaquant), mais dans ce cas, le chemin de recherche des commandes sera non seulement très différent mais surtout un témoin gênant pour l'attaquant.

Une technique beaucoup plus fine est d'implémenter un cheval de Troie dans le fichier d'initialisation qui s'exécute alors à l'entrée dans le système et de façon invisible pour la victime, d'autant plus que l'auteur forcera une remise de ce fichier dans son état original après son exécution, ce qui rendra toute détection impossible.

Certains terminaux intelligents interprètent des séquences de caractères bien particulières comme si elles avaient été frappées au terminal par l'utilisateur. C'est notamment le cas du `!` interprété par le *mail* comme le début d'une commande shell, le shell est donc déclenché et il interprète le reste de la ligne comme une commande. Il est aisé d'imaginer les conséquences désastreuses que peut avoir l'envoi d'un message du type `!rm -r *` par un attaquant.

2.3.4. EVALUATION DES BESOINS

La deuxième étape de notre méthode d'analyse de la problématique de la sécurité consiste en une analyse des besoins exprimés par les principaux clients, à savoir les utilisateurs et l'administrateur .

2.3.4.1. LES BESOINS DES UTILISATEURS

Les besoins des utilisateurs en matière de sécurité peuvent être regroupés en trois grandes classes : l'entrée dans le système, la protection des données et l'information.

Le besoin essentiel et premier de l'utilisateur est d'éviter que toute autre personne, utilisateur ou non du système, puisse entrer dans son univers de travail. Ce besoin est important parce qu'il reste la base de la majorité des atteintes possibles à la sécurité, il est donc celui qui est confronté à la plus grande partie des risques évoqués précédemment. Il est clair qu'il s'agit d'un besoin pour lequel l'utilisateur peut intervenir de façon décisive, la gestion correcte et efficace de son mot de passe représente une part considérable dans les moyens nécessaires pour assurer ce besoin. Nous reviendrons en détails sur ce service que l'utilisateur se rend à lui-même dans le point consacré aux services et aux serveurs de sécurité.

La protection des données est généralement le besoin final de l'utilisateur. Ce besoin peut être en partie assuré même si le premier besoin n'est plus garanti. Des moyens de protection de la confidentialité des informations doivent être utilisés de manière efficace par l'utilisateur, son intervention dans la réalisation de ce besoin est donc tout aussi primordiale que pour le précédent. L'intégrité des données reste, quant à elle, totalement tributaire d'une intrusion dans l'environnement de l'utilisateur.

Le besoin en information de l'utilisateur est un besoin que l'on peut considérer comme transitoire en ce sens que sa nécessité n'a d'autre intérêt que de servir les premiers besoins évoqués. La connaissance des mesures de sécurité mises à sa disposition et des techniques pour bien les utiliser constitue le ciment de l'édifice que représente la sécurité aux

yeux de l'utilisateur. Cette connaissance ne peut s'acquérir que par la pratique et l'expérience ou par la formation, d'où un besoin de formation découlant du besoin d'information.

2.3.4.1. LES BESOINS DE L'ADMINISTRATEUR

Les besoins de l'administrateur d'un système Unix sont doubles. D'un côté il peut être considéré comme un utilisateur du système et ses besoins sont similaires à ceux évoqués pour tout autre utilisateur mais il doit aussi être considéré comme un gestionnaire au sens strict du terme. Ce rôle lui impose alors des besoins très spécifiques.

Le besoin primordial, immédiatement évoqué par tout administrateur de système, est d'éviter qu'une autre personne ne puisse s'octroyer ses privilèges particuliers et mettre par là-même en péril la sécurité de tous les autres utilisateurs.

Une nécessité qui peut être considérée comme le pendant du besoin d'information des utilisateurs, est le besoin de contrôle de l'administrateur. Ce contrôle doit s'exercer sur les mesures et les techniques appliquées par les utilisateurs et il peut se réaliser au moyen d'outils d'audit informant le gestionnaire du niveau de sécurité actuel du système.

L'administrateur, de par son rôle, doit assurer aux utilisateurs du système son fonctionnement normal. Cette continuité des services constitue un troisième besoin évoqué par les administrateurs de système en matière de sécurité. Si ce besoin évoque l'idée de la fiabilité, que nous n'abordons pas dans ce mémoire, il fait également référence aux services assurés par le système pour la protection des informations.

2.4. SERVICES ET SERVEURS DE SECURITE

La deuxième étape de la méthodologie choisie consiste en une étude des services de sécurité fournis par les principaux serveurs. Ceux-ci sont, dans le cadre du problème posé, au nombre de quatre : le système d'exploitation, les outils logiciels, les utilisateurs et l'administrateur du système.

2.4.1. LE SYSTEME D'EXPLOITATION : UNIX

Le système Unix est un environnement reconnu pour son ouverture et son accessibilité, encourageant par là-même l'utilisateur à partager ses informations avec certains, voir tous les autres usagers.

Cette caractéristique d'ouverture n'implique pas un désintérêt pour la sécurité. Contrairement à la croyance populaire, le système Unix est fondamentalement sécurisé de par sa logique de conception. En effet, ce système d'exploitation est un gigantesque édifice composé de briques de base mises à la disposition de l'administrateur du système et même des utilisateurs, afin de leur permettre de spécifier le niveau de sécurité qui leur semble nécessaire.

Nous allons dans ce point présenter les outils offerts par Unix, tant aux utilisateurs qu'à l'administrateur, pour contribuer à l'établissement d'un système sûr.

2.4.1.1. SERVICES AUX UTILISATEURS

2.4.1.1.1. Le mot de passe

Comme dans beaucoup d'autres systèmes multiutilisateurs, l'identification de l'utilisateur est assurée par le *username* et l'authentification par un *password*. L'utilisation d'une telle technique suscite plusieurs problèmes et particulièrement dans le cas de Unix.

Le mot de passe de chaque utilisateur, afin qu'il soit disponible pour le système, est stocké dans un fichier appelé */etc/passwd*. Ce fichier lisible par tout utilisateur, est bien sûr protégé en écriture et la partie contenant le mot de passe est chiffrée, ce qui rend son utilisation impossible même pour le

superutilisateur. Un exemple typique du format d'une ligne de ce fichier est le suivant :

```
jean:ke6Ce;éb90j.5vZicy:116:116:info:/usr/users/students/jean:/bin/sh
```

Le premier champs représente le *username* , suivi du mot de passe chiffré, de deux nombres représentant le *user id* (UID) et le *group id* (GID). Le user-id est un nombre entier par lequel le système reconnaît l'utilisateur, le group-id est le numéro d'identification d'une classe d'utilisateurs à laquelle l'utilisateur appartient. Le champs *info* constitue une zone libre permettant d'ajouter des informations quelconques sur l'utilisateur (nom, adresse, fonction,...). Viennent enfin deux champs représentant le chemin du répertoire courant de l'utilisateur et le chemin du répertoire du *shell* où se trouvent les commandes Unix.

Le système compare donc la version chiffrée du mot de passe introduit par l'utilisateur avec celui qu'il possède dans le fichier susmentionné. Le fichier n'étant pas accessible en écriture, il est impossible de modifier la version chiffrée. Le service pour changer le mot de passe est offert par une commande Unix : *passwd* . La suppression de l'écho lors de la frappe du mot de passe, tant lors de la procédure d'entrée que lors de la procédure de modification, assure une partie de la confidentialité de l'interaction.

Cet outil reste évidemment tributaire d'une saine gestion par l'utilisateur. Nous reviendrons à ce problème dans le point considérant l'utilisateur sous l'angle du fournisseur de services de sécurité.

2.4.1.1.2. Les groupes

Au moment de l'entrée dans le système, l'utilisateur est associé à un groupe. Les nouveaux groupes sont créés uniquement par l'administrateur du système. Ils sont utilisés par les utilisateurs qui appartiennent à un même projet et qui désirent que les accès à leurs fichiers soient limités aux membres du groupe.

Le fichier */etc/group* contient des informations relatives au groupe qui se présentent sous la forme suivante:

a68: : 3: srb, a68

Le premier champ contient le nom du groupe, le champ suivant contient le mot de passe éventuel du groupe, le troisième champ représente le numéro de groupe et le dernier champ contient la liste des noms des utilisateurs qui appartiennent au groupe.

S'il fait également partie d'un autre groupe, l'utilisateur peut au cours d'une session changer de groupe courant en utilisant la commande *newgrp*. Le groupe associé au processus sera alors modifié en fonction du contenu du fichier */etc/group*. En effet, deux cas sont possibles : sans ou avec mot de passe. Dans le premier cas, seuls les utilisateurs explicitement nommés dans le fichier sont alors autorisés à exécuter la commande. Dans le deuxième cas peuvent avoir accès au groupe et à ses permissions les utilisateurs explicitement nommés dans le fichier (et ce sans donner de mot de passe) et les autres utilisateurs capables de donner le mot de passe.

2.4.1.1.3. La protection des fichiers

Tout système multiutilisateur et multitâche est doté d'un système de protection des fichiers. Dans le cas de UNIX, le système distingue pour chaque fichier trois niveaux de droits d'accès :

- celui du propriétaire du fichier
- celui du groupe
- celui de tous les autres utilisateurs

A chacun de ces trois niveaux, le propriétaire du fichier peut assigner trois modes d'autorisation :

- lecture (*read*)
- écriture (*write*)
- exécution (*exécution*)

Chaque fichier a une série de permissions qui lui sont accordées et qui déterminent qui peut faire quoi dans le fichier.

Chaque système UNIX possède en outre un super-utilisateur (super-user) qui peut lire et modifier chaque fichier du système.

On peut obtenir l'ensemble des renseignements à propos d'un fichier ou d'un répertoire en utilisant la commande `ls` avec l'option `-l` :

```
-rwxr-xr-x jean 160 sep 20 1989 abc  
-rwx----- jean 2000 oct 30 1989 xyz.
```

Pour la première ligne: le premier caractère indique le type du fichier : “-” pour un fichier ordinaire, “d” pour un répertoire. La chaîne `rwxr-xr-x` représente la protection du fichier. Les trois premiers caractères `rwx` indiquent les droits d'accès du propriétaire, les trois suivants ceux du groupe que le propriétaire a désigné et les trois derniers indiquent les droits d'accès de tous les autres utilisateurs .

Dans cet exemple, le propriétaire peut lire, écrire et exécuter le fichier, les membres du groupe et les autres utilisateurs peuvent le lire et l'exécuter. Le champ suivant: “jean” représente le numéro d'identification (user-id) du propriétaire du fichier. Le champ suivant indique la taille du fichier en octets. La date et l'heure sont celles de la dernière modification (écriture) du fichier. On trouve ensuite le nom du fichier en dernière position.

2.4.1.1.4. La protection des répertoires

Les répertoires disposent également d'un mécanisme de contrôle d'accès travaillant de façon similaire à celui des fichiers.

Certaines différences dans l'interprétation de l'état des permissions `r`, `w` et `x` sont toutefois importantes à relever. En résumé, on peut dire que la permission `r` permet d'exécuter une commande `ls` sur le répertoire, la permission `w` d'ajouter ou de retirer un fichier du répertoire et la permission `x` d'exécuter la commande `cd` ou d'insérer le répertoire dans un chemin plus important . Si on réfléchit à ce qu'est un répertoire, on peut plus aisément comprendre ces permissions. Le répertoire est un fichier contenant certaines informations sur plusieurs fichiers, il est donc logique que `ls` représente une lecture du fichier et que le fait d'ajouter ou de retirer un fichier d'un répertoire soit considéré comme

une opération d'écriture. La permission d'exécution, n'ayant pas d'utilité fonctionnelle pour un répertoire, est utilisée pour activer ou désactiver la permission de recherche (*search permission*) . Pour atteindre un fichier donné, il faut que cet attribut soit activé pour tous les répertoires se trouvant dans le chemin d'accès au fichier en question.

La liste suivante donne un aperçu des combinaisons possibles et utiles pour protéger l'accès à ses répertoires :

<i>drwxrwxrwx</i>	<i>1 dede students 512 Apr 01 18:34 tous</i>
<i>drwxr-xr-x</i>	<i>1 dede students 512 Apr 01 18:34 moi</i>
<i>drwxrwxr-x</i>	<i>1 dede students 512 Apr 01 18:34 groupe</i>
<i>drwxrwxrw-</i>	<i>1 dede students 512 Apr 01 18:34 groupe1</i>
<i>drwxrwxr--</i>	<i>1 dede students 512 Apr 01 18:34 groupe2</i>
<i>drwx-----</i>	<i>1 dede students 512 Apr 01 18:34 moi seul</i>
<i>d--x--x--x</i>	<i>1 dede students 512 Apr 01 18:34 personne</i>
<i>drwx---rwx</i>	<i>1 dede students 512 Apr 01 18:34 autres</i>

Le répertoire *tous* autorise tous les utilisateurs à créer et enlever des fichiers qu'il contient. Le répertoire *Moi* permet seulement au propriétaire de créer ou effacer des fichiers, les membres du groupe et les autres utilisateurs ayant les mêmes droits, à savoir un droit de listage du répertoire et un droit de recherche. *Groupe* est identique au répertoire *Moi* sauf qu'il donne les mêmes droits aux membres du groupe qu'au propriétaire. Le répertoire *groupe1* octroie tous les droits au propriétaire et au groupe mais n'autorise les autres utilisateurs qu'à lister le répertoire, le droit d'écriture étant inopérant sans le droit de recherche. *Groupe2* a un effet totalement identique à *groupe1* . Le répertoire *moi seul* n'est accessible que par le propriétaire, les autres utilisateurs ne peuvent accéder à aucun de ses fichiers. Dans le répertoire *personne* , la suppression et le changement de nom sont interdits à quiconque, de plus personne ne peut lister les fichiers. Cependant si quelqu'un connaît le nom d'un fichier il peut l'atteindre et bénéficier des permissions d'accès propres à ce fichier. Enfin le répertoire *autres* montre que l'appartenance à un groupe peut être considérée comme restrictive par rapport aux autres utilisateurs.

2.4.1.1.5. La modification des protections

La protection d'un fichier peut être modifiée par le propriétaire ou par le super-utilisateur à l'aide de la commande *chmod*. La protection des fichiers y est décrite par un nombre octal. Ce nombre est déterminé par la somme du droit de lecture (valeur 4), du droit d'écriture (valeur 2) et du droit d'exécution (valeur 1).

Par exemple, *chmod 751 run* va modifier la protection du fichier *run* de la manière suivante: *rwX r-x --x* autorisant un accès sans restriction pour le propriétaire, un accès en lecture et en écriture pour les membres du groupe et un accès en exécution pour les autres.

La commande *chmod* permet également une description relative des droits d'accès. Par exemples :

<i>u-r</i>	supprime l'accès en lecture pour le propriétaire
<i>g+w</i>	ajoute l'accès en écriture pour le groupe
<i>o=x</i>	établi l'accès en exécution pour les autres utilisateurs

2.4.1.1.6. Les protections par défaut

Les permissions par défaut lors de la création d'un fichier ou d'un répertoire sont déterminées par la commande *umask*. Il est utile de placer cette commande dans le fichier *.profile*, celui-ci étant exécuté à chaque entrée dans le système. Cette commande n'empêche nullement de modifier les protections après la création au moyen de la commande *chmod* vue en détail dans le chapitre traitant de la gestion utilisateur. Le principe de fonctionnement de cette commande est l'opposé de *chmod*, elle stipule les permissions qui ne doivent PAS être accordées lors de la création du fichier ou du répertoire. Ainsi *umask 0* est équivalent à une absence de restrictions, les nouveaux fichiers créés auront tous la protection *rwX rwX*. *umask 22* est la commande par défaut utilisée sur de nombreuses versions du système, elle indique une protection maximale égale à *rwX r-x r-x*. *umask 77* fixera la protection par défaut à *rwX --- ---*.

2.4.1.1.7. Le SUID et SGID

Certains fichiers exécutables possèdent en outre les attributs complémentaires "S" ou "T" qui viennent s'ajouter à la suite des attributs de droits d'accès. Ils correspondent respectivement au "set-user-id" (SUID) ou "set-group-id" (SGID) et au "sticky bit".

Ces attributs ne sont utiles que pour des fichiers contenant un code programme exécutable. Ils permettent de donner temporairement à un processus des droits spécifiques, différents de ses droits propres initiaux (correspondant aux droits de l'utilisateur du processus).

Lorsqu'un processus est en cours d'exécution, le système lui attribue deux identificateurs : l'identificateur réel toujours égal à celui de l'utilisateur du processus et l'identificateur effectif qui peut être temporairement différent. Le système crée deux autres identificateurs similaires, mais relatifs au groupe de l'utilisateur. Généralement, les identificateurs effectifs sont identiques aux identificateurs réels sauf si le processus doit exécuter le code contenu dans un fichier sans en avoir les droits d'exécution ordinaires (x), mais dont le bit d'attribut "set-user-id ou"set-group-id" est positionné. Dans ce cas, le processus peut quand même exécuter le code contenu dans ce fichier avec l'identificateur effectif du processus égal à l'identificateur du propriétaire du fichier. Le processus possède alors des droits égaux à ceux du propriétaire du fichier tant qu'il exécute le code en question.

Dès qu'il doit exécuter le code contenu dans un autre fichier, l'identificateur effectif d'utilisateur (ou du groupe d'utilisateur) redevient égal à l'identificateur réel. Le processus continue alors à s'exécuter avec ses droits initiaux.

Le set-user -id est souvent positionné pour permettre la mise-à-jour d'un fichier de données avec des informations fournies par l'utilisateur d'un processus, sans que ce dernier puisse accéder directement au fichier de données concerné. La mise à jour pourra se faire par un programme dont le set-user-id est positionné. L'utilisateur ne pourra donc mettre à jour les données que dans les limites permises par le programme.

Le "sticky bit" permet d'accélérer l'accès au fichier dans lequel il est positionné. Si les données, généralement du code exécutable, doivent libérer temporairement la mémoire primaire, le "swapper" (processus échangeur) les range dans sa zone d'échange en 1 bloc unique continu, au lieu de le remettre dans le répertoire d'origine dans des blocs discontinus. A chaque demande, les accès sont plus rapides car le contrôleur de disque provoque des mouvements mécaniques de la tête de lecture très réduits .

2.4.1.1.8. Le chiffrement des fichiers

Pour garantir encore mieux la confidentialité des données importantes, Unix offre la possibilité à l'utilisateur de les chiffrer au moyen de la commande *crypt* . Cette commande utilise une clé pour chiffrer l'entrée standard en un résultat lisible envoyé sur la sortie standard. Dans l'exemple suivant, le chiffrement du fichier *perf.jc* au moyen de la clé *xyz* dans le fichier chiffré *perf.ch* se fait au moyen de la commande

```
crypt xyz < perf.jc > perf.ch
```

Pour récupérer le contenu lisible du fichier, il suffit d'exécuter la même commande en spécifiant la même clé que pour le chiffrement, ce qui donne dans notre exemple

```
crypt xyz < perf.ch > perf.jc
```

L'utilisation de la commande *crypt* propose une variante plus intéressante au niveau sécurité. Dans cette variante, on ne spécifie pas la clé comme argument de la commande, cette clé sera demandée par le système sous la forme :

```
crypt < perf.jc > perf.ch  
enter key : xyz
```

Cette façon de procéder revêt deux avantages. Dans ce cas, la clé n'est pas affichée à l'écran, ce qui garantit tout comme pour le mot de passe une plus grande confidentialité. De plus, dans la

première variante, il était possible pour un autre utilisateur de connaître cette clé au moyen de la commande *ps -f* , celle-ci imprimant les informations sur les processus en cours avec le nom de la commande et tous ses arguments, la clé est donc lisible entre le moment où la commande est entrée et celui où la procédure de chiffrement commence. L'intérêt du chiffrement des données va de pair avec l'obligation de supprimer les versions lisibles de ces mêmes données et d'employer de "bonnes" clés, c'est-à-dire difficiles à déduire et faciles à retenir. Les remarques faites au sujet du mot de passe sont donc applicables dans ce cas.

Pour être certain que seules les versions chiffrées existent sur disque, les éditeurs du système Unix (*ed* , *vi* et *emacs*) permettent grâce à l'option -x de ne travailler qu'avec des fichiers chiffrés.

Pour accroître l'efficacité du chiffrement, on peut utiliser la commande *pack* avant *crypt* . Cette commande permet de compresser les données dans un nouveau fichier suffixé .z tout en effaçant la version originale. Il n'est plus possible dans ce cas d'utiliser l'option -x des éditeurs de Unix.

La commande *makekey* , outre le fait qu'elle soit utilisée par le système pour le chiffrement du mot de passe et sa comparaison au moment de l'entrée dans le système, permet de diminuer considérablement la probabilité de voir une clé découverte.

Elle permet en effet, à partir d'une suite de dix caractères, de les chiffrer en une suite de treize caractères dont les deux premiers sont identiques aux deux derniers de la suite entrée en argument et représente un sélecteur (choisi dans l'ensemble des chiffres et des lettres) d'une des méthodes cryptographiques de l'algorithme DES.

<i>/usr/lib/makekey</i>	ou	<i>/usr/lib/makekey</i>
<i>cryptage7Y</i>		<i>cryptage</i>
<i>7YiebRséam\$3p</i>		<i>7Y</i>
		<i>7YiebRséam\$3p</i>

Il est néanmoins utile de signaler que la clé donnée par *makekey* est affichée à l'écran et que la commande *ps -f* permet de lire tous les arguments du processus exécutant la commande *makekey* et donc la clé.

2.4.1.2. SERVICES A L'ADMINISTRATEUR

Outre les services classiques offerts par le système à l'administrateur en tant qu'utilisateur normal des ressources, Unix met à sa disposition un certain nombre d'outils spécifiques à sa fonction.

Le premier service que le système fournit à l'administrateur, est de lui donner un statut particulier appelé superutilisateur (*super user*). Ce statut est la base sur laquelle reposent les autres services. Plusieurs commandes d'administration ne peuvent être exécutées que par le superutilisateur. Unix lui permet de lire et d'écrire tous les fichiers sans tenir compte des permissions et d'exécuter tous les programmes. Son nom d'utilisateur est toujours *root* et son numéro d'identification 0.

Nous allons dresser une liste non exhaustive des privilèges offerts à ce superutilisateur par Unix.

2.4.1.2.1. Les fichiers et répertoires système

Certains répertoires et fichiers jouent un rôle particulier prépondérant dans le bon fonctionnement du système. Plusieurs d'entre eux ont un impact important sur la sécurité.

Les programmes Unix peuvent être considérés comme indépendants des périphériques, en ce sens que l'accès à ces derniers se fait par le système de fichiers classique. Pour accéder à un périphérique, un programme ouvre simplement le fichier lui correspondant et l'utilise comme un fichier Unix normal. D'un point de vue sécurité, il s'agit d'un service intéressant, puisque toutes les entrée/sortie de et vers les différents périphériques se font à travers un très petit nombre de canaux, les fichiers de périphériques. Si les permissions d'accès aux partitions disque sont configurées proprement, les utilisateurs ne peuvent accéder au disque que via le système de fichier standard possédant son propre mécanisme de

sécurité. Malheureusement, si les permissions d'accès aux partitions disque ne sont pas configurées correctement, tout utilisateur sera capable d'écrire un programme pouvant lire chaque fichier de la partition disque en lisant simplement le **i-node** et ensuite en lisant les blocs dans l'ordre dans lequel ils apparaissent sur la liste des adresses disque.

La commande *find* utilisée pour traverser une hiérarchie de répertoires et exécuter certaines opérations sur tous les fichiers qui y sont contenus, peut rendre un service particulier à l'administrateur dans le cadre de ces fichiers périphériques. L'option *-type* de cette commande suivie de la lettre *b* permet de sélectionner les fichiers spéciaux (*block special files*) dans la hiérarchie des utilisateurs. Etant donné qu'un utilisateur ordinaire ne devrait jamais posséder ce genre de fichiers, une action s'impose (enlever le fichier, réprimander voir exclure l'utilisateur, ...).

Le fichier *etc/passwd* est un des fichiers clé dans la sécurité de Unix. Comme nous l'avons vu précédemment, il est utilisé pour vérifier le mot de passe des utilisateurs qui entrent dans le système. Il est naturellement accessible en écriture uniquement par le superutilisateur.

Le format de ce fichier permet à l'administrateur d'exiger que les utilisateurs changent périodiquement leur mot de passe. Si on prend attention au format de ce fichier, on peut constater que certains des mots de passe chiffrés contiennent une virgule, suivie d'un certain nombre de caractères et terminé par le double point séparateur de champs.

```
root:xyDfcc5!lsA9w,M.y8:0:0:administrateur:./bin/sh
console:kl5çvdag8pqxi,M.y8:0:0:administrateur:./bin/sh
jean:l9v%zvEnK3!mw:0:0:Dupond J:/usr/users/jean:/bin/sh
marc:nC4mltV7?rbll,M.z8:0:0:Martin M:/usr/users/marc:/bin/sh
serge:bGe1a£pmOny6F,./:0:0:Godin S:/usr/users/serge:/bin/sh
```

Les mots de passe chiffrés des utilisateurs *root*, *console* et *marc* contiennent quatre caractères après la virgule, *serge* deux et *marc* aucune virgule. Le premier caractère après la virgule

détermine le nombre maximum de semaines de validité du mot de passe, le second le nombre minimum de semaines qui doivent s'écouler avant que le mot de passe ne puisse être changé par l'utilisateur. Les caractères restant indiquent quand le mot de passe a été changé dernièrement. L'interprétation des caractères n'est possible que sur base d'un tableau de correspondances.

caractère	nombre	caractère	nombre
.	0	B	13
/	1	C	14
0	2	D	15
1	3	E	16
2	4
3	5	Y	36
4	6	Z	37
5	7	a	38
6	8	b	39
7	9	c	40
8	10
9	11	y	62
A	12	z	63

Pour reprendre l'exemple de l'utilisateur Marc, le *M* situé après la virgule signifie que le mot de passe doit être changé toutes les 25 semaines au moins. Le point suivant le *M* signifie que le mot de passe peut être changé aussi souvent que l'utilisateur le désire. Le *z8* indique quand le mot de passe a été changé pour la dernière fois, cette information est utilisée lorsque l'utilisateur entre dans le système afin d'être comparée avec le temps d'expiration prévu. Les deux premiers caractères sont gérés par l'administrateur, alors que les deux derniers caractères sont positionnés par la commande *passwd* lorsque l'utilisateur change le mot de passe. Deux configurations spéciales de ce format méritent une attention particulière. Premièrement lorsque le nombre maximal de semaines (premier caractère) est inférieur au minimum (deuxième caractère). Dans ce cas, il n'est pas permis à l'utilisateur de changer son mot de passe, seul le superutilisateur peut le faire, c'est le cas de l'utilisateur *Serge* dans notre exemple. Deuxièmement, lorsqu'aussi bien le

premier que le deuxième caractère sont un point. Dans ce cas, l'utilisateur est obligé de changer son mot de passe à sa prochaine entrée dans le système. Après quoi, les deux points sont retirés par le système et l'utilisateur n'est jamais plus obligé de changer son mot de passe. Un problème a toutefois été soulevé à l'encontre de ce genre de technique [WOO,51], l'obligation de changer de mot de passe lors d'une entrée dans le système impose à l'utilisateur un choix rapide et donc moins réfléchi du mot de passe, ce qui est contraire à sa saine gestion. Il est à noter que la commande *su* ne contrôle pas l'expiration du mot de passe.

Le système Unix permet d'avoir deux utilisateurs ayant le même numéro d'identification, leur octroyant les mêmes droits d'accès sur les fichiers. Généralement, le nom d'utilisateur indiqué par les commandes du type *ls*, *ps* sera celui rencontré en premier lieu dans le fichier */etc/passwd*.

Le fichier */etc/group*, comme nous l'avons vu précédemment, contient des informations sur les groupes. Chaque numéro d'identification du groupe (GID) du fichier */etc/passwd* a une entrée correspondante dans le fichier */etc/group*. Ce fichier rend la gestion des groupes plus facile à l'administrateur, mais n'est pas nécessaire pour que le principe des groupes fonctionne. En effet, le système utilise les numéros d'identification de l'utilisateur et du groupe provenant du fichier */etc/passwd* pour déterminer les droits d'accès aux fichiers, et ce même si le fichier */etc/group* n'existe pas.

Il y a encore beaucoup de fichiers sur le système Unix qui ont une fonction particulière et qui ne peuvent jamais être accessibles en écriture pour les utilisateurs. S'y retrouvent notamment les fichiers */etc/inittab*, */usr/lib/crontab*, toutes les commandes dans */bin* et */usr/bin* et plusieurs répertoires tels que */usr*, */usr/lib* et bien d'autres. Outre les permissions sur les fichiers, Unix offre à l'administrateur un système de protection des répertoires les contenant identique à celui décrit dans les services offerts à l'utilisateur.

2.4.1.2.2. La gestion des utilisateurs

Le système Unix permet au superutilisateur d'ajouter ou de retirer un utilisateur. L'ajout d'un utilisateur doit se réaliser en trois étapes, la création du mot de passe, la création d'un répertoire et la création d'une appartenance à un groupe éventuel. Le retrait d'un utilisateur comporte trois phases similaires : la suppression du mot de passe, le retrait du ou des groupes au(x)quel(s) il appartient et l'effacement de son répertoire et de tous ses fichiers.

2.4.1.2.3. Les commandes système

Il existe de nombreuses commandes disponibles uniquement pour le gestionnaire ou pour tous les utilisateurs (voir annexe ?) du système traitant notamment de la comptabilisation, de la gestion des ressources, des priorités, etc. Parmi les commandes ayant une utilité pour la gestion de la sécurité par l'administrateur, nous en avons relevé trois, *ncheck* , *mount* et *find* , cette dernière étant accessible par tous les utilisateurs.

La commande *ncheck* est utilisée pour contrôler le système de fichiers. Elle permet notamment à l'administrateur de lister tous les numéros d'i-node et le fichier qui leur est associé pour une partition du disque. Elle donne accès avec l'option *-i* aux fichiers à partir de leur numéro d'i-node et permet ainsi de détecter les fichiers identiques mais de noms différents construits au moyen de lien.

La commande */etc/mount* a pour but d'installer et de désinstaller un système de fichiers sur une partition, mais peut être utilisée sans argument pour obtenir la liste des répertoires où le système de fichiers est installé.

La commande *find* est utilisée, comme nous l'avons stipulé précédemment, pour traverser une hiérarchie de répertoires et exécuter certaines opérations sur les fichiers qu'ils contiennent. Elle est utilisée avec le nom du répertoire qui sert de point de départ et une ou plusieurs options qui indiquent les opérations à réaliser sur les fichiers. L'option *-print* affiche tous les fichiers dans la hiérarchie

sous le répertoire courant. On peut aussi donner en option le nom de l'utilisateur pour réduire la sélection des fichiers à ceux possédés par cet utilisateur. La commande *find*, lorsqu'elle est exécutée par *root*, passe outre des protections des répertoires scannés, ce qui lui donne un caractère particulier pour l'administrateur, malgré son accessibilité à tout utilisateur. L'option *-perm mode* recherche les fichiers avec la protection *mode* en octal. L'option *-type*, suivie d'une lettre spécifiant le type de fichier recherché, permet de sélectionner les fichiers à trouver sur cette base (*b* pour les fichiers spéciaux de bloc et *c* pour les fichiers spéciaux de caractères). L'option *-exec commande* ; permet à l'administrateur d'exécuter commande sur chaque fichier trouvé.

2.4.2. LES UTILISATEURS

Les services que peuvent rendre les utilisateurs s'adressent généralement à eux-mêmes, parfois à l'administrateur en lui facilitant sa tâche de gestion et de contrôle par un respect des techniques et des principes de base. L'utilisateur peut agir dans deux domaines ayant un impact important sur la sécurité : la gestion de son mot de passe, la protection de ses données.

2.4.2.1. LA GESTION DU MOT DE PASSE

Les précautions prises par le système ne suffisent pas à garantir l'efficacité de l'utilisation du mot de passe. L'utilisateur doit à son tour prendre certaines précautions en ce qui concerne son choix et sa durée de validité.

2.4.2.1.1. Choix du mot de passe

Le choix du mot de passe est d'une importance capitale, d'autant que certaines versions de Unix n'imposent aucune restriction quant à son format. La règle est de ne pas utiliser un mot de passe évident à déduire, il est donc à proscrire d'utiliser noms, prénoms, initiales, date de naissance. Il est aussi recommandé d'éviter les mots de passe déductibles d'un objet dans la pièce. Le

fait de suffixer ou de préfixer ce type de mot de passe par un chiffre ou un caractère spécial n'augmente pas de façon significative sa valeur.

Comment choisir un "bon" mot de passe ?

Sa longueur ne peut impérativement est inférieure à six caractères. Certaines versions garantissent d'ailleurs cette condition. Il ne peut être basé sur une information personnelle ou être un mot du dictionnaire courant. Une excellent solution consiste à utiliser les initiales des mots composant une phrase, sans toutefois que celle-ci ne soit trop courant (réplique connue du théâtre, ...).

Une technique souvent proposée est d'effectuer un mélange des lettres de deux mots courants et qui ne sont pas liés. Une autre solution très efficace est d'utiliser une série de caractères pris au hasard. Si cette approche s'avère efficace, elle s'avère néanmoins risquée à utiliser vu la difficulté de rétention de ce genre de mot de passe, surtout si sa longueur est importante.

Il est impératif qu'il ne contienne pas que des caractères alphabétiques. Il est recommandé d'insérer un ou plusieurs caractères spéciaux (chiffres, signes de ponctuation, ...). Cela semble être une évidence, mais il nous paraît utile de le rappeler, le mot de passe ne peut être écrit ou stocké dans un endroit accessible.

Le mot de passe doit être tapé dans trois cas : l'entrée dans le système, le changement de mot de passe et l'activation de la commande *su* . Si le mot de passe n'est pas affiché à l'écran, il est toutefois recommandé de vérifier que personne ne peut voir sur le clavier la frappe de celui-ci. Certaines techniques permettent de reconnaître la signature sonore des touches du clavier ou de capturer les caractères transmis entre le clavier et l'ordinateur. Si le niveau de sécurité l'exige, des mesures doivent être prises contre de tels procédés.

2.4.2.1.2. Durée de validité du mot de passe

La première règle à respecter est de proscrire l'utilisation d'un même mot de passe sur différentes machines. C'est spécialement d'application si les différentes configurations se situent à des niveaux de sécurité différents.

Outre le fait que l'administrateur peut forcer le changement du mot de passe, la seconde règle consiste à se fixer une discipline de modification périodique du mot de passe. L'intervalle de validité est une notion subjective faisant intervenir plusieurs facteurs tels que valeur des informations, niveau de sécurité de la configuration, etc . Il est généralement admis qu'une durée d'utilisation de six mois est un maximum à ne pas dépasser, la solution idéale étant le changement à chaque entrée dans le système. Une telle pratique aurait néanmoins pour conséquence de rendre difficile le choix d'un bon mot de passe et d'augmenter la probabilité de ne plus s'en souvenir.

2.4.2.2. LA PROTECTION DES DONNEES

L'utilisateur peut assurer la protection de ses données au moyen de deux techniques : la protection des fichiers et répertoires et le chiffrement des données.

2.4.2.2.1. La protection des fichiers et répertoires

L'utilisateur doit se servir au maximum des trois permissions d'accès aux fichiers. La permission d'écriture ne peut être octroyée qu'à soi-même ou éventuellement au groupe si celui-ci est bien défini et que le travail pour lequel il existe nécessite des modifications des données, mais en aucun cas cette permission ne doit être accordée aux autres utilisateurs. Cette règle permet de garantir, si le répertoire est bien protégé, l'intégrité des informations.

La confidentialité, si elle est nécessaire, peut être assurée par les mêmes règles appliquées à la permission de lecture.

Les répertoires contenant les fichiers protégés doivent aussi faire l'objet d'une réduction des permissions d'accès, sans quoi les protections placées sur les fichiers seraient totalement hypothéquées. Afin de garantir l'intégrité des fichiers qu'il contient, un répertoire ne peut être accessible ni en écriture, ni en exécution pour les autres utilisateurs. Ces permissions peuvent éventuellement être élargies pour les membres du groupe. La confidentialité sera garantie si le répertoire est protégé en lecture et en exécution, ce qui empêchera tout autre utilisateur de prendre connaissance des fichiers et de leur contenu. Cette restriction d'accès peut être plus faible pour les membres du groupe en leur donnant une permission de recherche dans les répertoires sans pour autant leur laisser un droit de lecture. Cette combinaison permet une lecture des fichiers dont les autres membres du groupe connaissent le nom et ne leur octroie pas la possibilité de lister tous les fichiers.

Afin d'éviter toute négligence dans la protection des nouveaux fichiers et répertoires, il est conseillé à l'utilisateur de donner une protection par défaut maximale, à savoir *umask 077*.

Les fichiers d'initialisation *.profile* et *.exerc*, étant des maillons sensibles dans la chaîne de la sécurité, doivent faire l'objet d'une protection maximale, même vis-à-vis des membres du groupe. Il est donc impératif de respecter la combinaison de permissions d'accès de l'exemple suivant tant pour le fichier *.profile* que pour le répertoire qui le contient :

```
drwxr-x---      1 dedo students 512 Apr 23 15:38 dedo
-rwx --- ---    1 dedo students 127 Apr 5  9:12 .profile
```

Un autre précaution est de demander dans le *.profile* un affichage à l'écran de la date et de l'heure de la dernière entrée dans le système.

Afin d'éviter de voir ses données détruites alors qu'elles étaient correctement protégées, il est utile de se prémunir contre les attaques utilisant la messagerie comme véhicule d'une commande dangereuse. Nous avons vu précédemment que la commande *write* permettait l'envoi d'un court message à l'écran, et qu'une utilisation

plus dangereuse pouvait engendrer l'exécution d'une commande comme si elle avait été introduite par l'utilisateur (si le message est précédé d'un "!"). La seule solution pour éviter la réception d'un message contenant une telle séquence est d'utiliser la commande *msg n* dans son fichier *.profile*.

Un autre domaine où l'intervention de l'utilisateur nous paraît importante est le transfert d'informations. Le transfert des données implique l'utilisation de répertoires d'accès libre, servant de "dépôt provisoire", ces répertoires tels que */uucppublic/receiver* ne garantissent donc nullement la confidentialité et l'intégrité des informations transférées. Le mouvement d'informations entre hôtes évoque automatiquement le passage par un réseau de communication constituant un point sensible dans la sécurité. La confidentialité des informations qui circule sur le support n'est donc pas garantie avec les services que nous avons décrits dans ce point.

Le système Unix utilise les répertoires */tmp* et */usr/tmp* pour y placer les fichiers temporaires issus de processus tels que le tri ou la compilation C. Ces répertoires étant accessibles en recherche, lecture et écriture par tout utilisateur, certains problèmes de confidentialité et d'intégrité peuvent se poser. Tout d'abord, le fait que la commande *sort* sauve une version provisoire d'un fichier à trier dans ce répertoire, peut avoir des conséquences fâcheuses si ce fichier contient des informations sensibles. Un deuxième problème peut se poser pour un programmeur qui utilise ces répertoires pour y placer ses propres fichiers temporaires dans le but d'utiliser moins d'espace disque propre. En effet, ce type de pratique peut avoir pour conséquence qu'un autre utilisateur puisse non seulement lire ces fichiers, mais aussi les modifier par remplacement puisque ce répertoire est accessible pour toute opération. L'utilisateur peut se rendre deux services dans ce contexte, tout d'abord éviter d'utiliser ces répertoires du système pour y placer ses propres fichiers temporaires et ensuite "nettoyer" régulièrement ces répertoires systèmes des fichiers lui appartenant et étant issus d'une application système telle que le tri ou la compilation C.

2.4.2.2.2. Le chiffrement des fichiers

Comme nous l'avons expliqué en détails, le système offrent plusieurs services de chiffrement à l'utilisateur. Ce dernier a les moyens d'améliorer ces services de base en les utilisant de manière efficace et opportune. L'usage de la commande *crypt* précédée éventuellement de la commande *pack* et de la commande *makekey* pour choisir une bonne clé s'impose dans certains cas et est conseillée dans d'autres. Cette technique constitue un excellent service pour le transfert de fichier, elle garanti l'illisibilité directe des informations circulant sur le réseau et transitant dans les répertoires "dépôts". Il peut constituer un outil idéal de protection des information contre l'administrateur, c'est la seule solution pour en garantir la confidentialité totale. Cette méthode peut également être employée pour garder secrètes des informations se trouvant dans un répertoire "partagé", c'est-à-dire accessibles par d'autres utilisateurs pour diverses raisons (les membres d'un groupe de travail par exemple). Cela constitue l'arme ultime cumulée à toutes les autres pour maintenir confidentielles des informations hautement sensibles ou personnelles. Une remarque est à faire concernant le choix de la clé de chiffrement, les conseils donnés pour la gestion du mot de passe sont pleinement d'application pour le choix d'une clef efficace. L'emploi de *makekey* diminue considérablement la probabilité de réussite d'une tentative d'attaque sur la clé.

Le chiffrement des fichiers est donc un moyen efficace de cacher une information importante soit à des intrus qui auraient franchi la barrière du mot de passe ou des permissions d'accès au fichier, soit à l'administrateur du système.

2.4.3. L'ADMINISTRATEUR

L'administrateur du système Unix a un double rôle en matière de sécurité. D'une part gérer la sécurité dans son acceptation stricte tant logique que physique, et d'autre part administrer la sécurité prise dans le sens de la fiabilité du système et des services qu'il rend à l'utilisateur. Ce second aspect de la mission de l'administrateur ne faisant pas l'objet de ce mémoire, nous nous attarderons uniquement sur une partie de son premier rôle, celle concernant la sécurité logique .

Cette partie de sa première mission peut être subdivisée à notre avis en trois fonctions bien particulières : Le contrôle externe, le contrôle interne et la formation des utilisateurs.

2.4.3.1. LE CONTROLE EXTERNE

Le contrôle externe regroupent les services que devrait fournir l'administrateur pour diminuer les risques d'accès non autorisés au système, et les risques découlant de cet accès, par des personnes externes , c'est-à-dire non connue comme utilisateur.

Le premier service que l'administrateur peut se rendre et rendre aux utilisateurs est la protection maximale du statut de superutilisateur. Les conseils évoqués pour le mot de passe d'un utilisateur classique sont doublement d'application pour le mot de passe du superutilisateur. Les conséquences d'une divulgation de ce dernier seraient catastrophiques vu qu'elles octroient des priorités maximales, touchant tous les utilisateurs, à celui qui a pu l'acquérir.

L'administrateur se doit de distribuer avec parcimonie les accès au système, cette mesure aura également un impact sur la seconde mission de l'administrateur, à savoir assurer la fiabilité aux utilisateurs. Cette sélection dans la distribution des accès facilite la fonction de contrôle interne que nous développerons au point suivant, elle diminue également les risques potentiels contre les données émanant des utilisateurs. La première solution pour une personne ayant des intentions malveillantes, avant même d'essayer de percer le système, est de demander à y être introduit légalement. La solution consiste donc à notre avis à analyser en

détails les motivations et les arguments avancés par le demandeur, la fonction de contrôle interne reprenant ensuite le suivi du respect de ces motivations.

La surveillance des accès au système, notamment les entrées dans le système à des heures anormales et/ou par des utilisateurs qui n'auraient pu le faire (car ils sont absents à l'étranger, ...), constitue le deuxième rôle de l'administrateur en matière de contrôle externe. La connaissance des points d'entrée possibles dans le système implique une connaissance préalable de la configuration exacte de l'environnement. Cette connaissance permet à l'administrateur de localiser plus facilement la provenance d'une entrée et d'accentuer la surveillance à cet endroit. Les différents points d'entrée dans le système sont assez facilement déductibles du schéma général de la configuration repris au chapitre 1, nous pouvons citer à titre d'exemple les stations de travail Vax, Sun, HP, Vax-VMS, les terminaux situés dans différents locaux, les micro-ordinateurs reliés au réseau soit par une carte Ethernet, soit par un logiciel tel que Kermit, les accès venant de l'extérieur par le réseau DCS-X25 ou par les lignes téléphoniques accessibles via le PABX,

La détermination du point d'entrée physique dans le système, telle que nous l'avons évoquée au paragraphe précédent, constitue une première localisation de l'intrus mais il nous paraît utile que l'administrateur puisse déterminer la localisation logique de l'intrus, c'est-à-dire le nom d'utilisateur qu'il a employé pour pénétrer le système. Cette localisation logique est essentielle pour la prise de mesures de réduction du risque, telles que l'obligation de changer le mot de passe pour l'utilisateur victime, le changement éventuel de son nom d'utilisateur,

Le fait d'analyser régulièrement l'activité réelle du réseau en comparaison avec l'activité attendue pour la période surveillée peut apporter des informations sur d'éventuelles variations témoignant d'une charge inhabituelle du réseau pour une période de référence. L'utilisation d'un outil de mesure de performances ou de charge du système peut donc s'avérer intéressant dans la lutte contre les accès non autorisés.

Le contrôle externe de la sécurité logique doit bien entendu être secondé par un contrôle d'accès physique aux différents points d'entrée

du système, contrôle qui ne fait pas l'objet de ce mémoire mais dont l'importance est primordiale sur la réduction de ce type de risque.

2.4.3.2. LE CONTROLE INTERNE

Le contrôle interne définit les services que doit fournir l'administrateur pour diminuer les risques "intérieurs" d'atteinte à la confidentialité et à l'intégrité des informations appartenant aux utilisateurs du système. Nous entendons par risques intérieurs, le fait qu'ils émanent d'un utilisateur régulier introduit dans le système ou d'une personne l'ayant au préalable pénétré illégalement.

L'administrateur peut dans cette fonction de contrôle interne fournir des services de sécurité aux utilisateurs mais aussi à lui-même. Nous ne distinguerons pas ouvertement ces deux catégories de services car elles ont une partie commune importante, nous nous contenterons donc de faire ressortir par service, le client auquel il s'adresse.

Le premier type de service qui doit émaner de cette fonction de contrôle a trait à la surveillance des ressources particulières du système que nous avons déjà développées dans le point consacré au système en tant que serveur de sécurité. Cette surveillance particulière de certaines ressources s'adresse tant aux utilisateurs qu'à l'administrateur lui-même. Nous allons aborder quelques ressources devant faire l'objet d'un intérêt particulier sans prétendre les avoir toutes abordées ni avoir considérer les plus importantes.

La surveillance particulière des partitions disques s'impose pour s'assurer qu'aucun utilisateur ne puisse accéder aux données autrement que via les fichiers de périphériques et donc resté soumis aux protections de ces fichiers ordinaires. L'accès aux partitions disques doit donc être minimal et la recherche d'une version copiée chez un utilisateur au moyen de la commande *find* peut s'avérer intéressante, cette présence serait significative de deux choses, la vulnérabilité de ce ou ces fichier(s) particulier(s) et la personnalité d'un des utilisateurs du système.

Le contrôle de certains fichiers système impose de se soucier du nombre de liens créés sur ces fichiers, la commande *ncheck* déjà évoquée peut permettre à l'administrateur de détecter ce type d'attaque.

La surveillance particulière de certains fichiers sensibles du système, tels que les fichiers */etc/passwd* ou */lib/crontab*, est une nécessité à laquelle l'administrateur doit s'astreindre. Une modification du fichier des mots de passe pourrait être réalisée si l'attaquant a pu pénétrer dans le système en tant que superutilisateur, il est donc utile pour l'administrateur de maintenir une copie (même sur papier) du fichier des mots de passe à chaque fois qu'il le modifie personnellement et de comparer cette copie régulièrement avec le fichier système.

La nécessité d'une protection maximale du fichier *.profile* du superutilisateur est d'autant plus importante que ce fichier constitue un point d'attaque particulièrement apprécié des amateurs du cheval de Troie.

L'administrateur se doit également de surveiller périodiquement ces ressources particulières notamment suite à des mises à jour ou à des installations de nouveaux logiciels qui sont généralement une occasion de substitution des permissions originales

Une deuxième famille de services inhérents à cette fonction de l'administrateur a trait à l'audit des mesures de sécurité prises par les utilisateurs eux-mêmes. Ce rôle de contrôle et d'évaluation permet de détecter les failles, les ouvertures béantes que certains utilisateurs laissent dans le système.

Un intérêt particulier doit être porté à l'aide à la gestion du mot de passe par les utilisateurs. Ce service reviendra également dans la mission de formation que doit remplir l'administrateur et que nous évoquerons ultérieurement. Le système Unix donne à l'administrateur les armes pour gérer la durée de validité d'un mot de passe au travers des caractères situés dans le mot de passe chiffré du fichier */etc/passwd* après la virgule. Cette possibilité a été expliquée en détail dans le point consacré aux services offerts par le système à l'administrateur. Ce dernier se doit donc d'utiliser cette faculté qui lui permet de s'assurer une diminution sensible des risques par une aine gestion du mot de passe de l'utilisateur. Si l'utilisation de cette possibilité peut se faire de manière manuelle, elle sera encore plus rentable si elle fait l'objet d'un ensemble automatisé, nous reviendrons sur cette perspectives dans le point que nous consacrerons aux outils serveurs de sécurité.

L'évaluation des protections des fichiers et répertoires des utilisateurs constitue également une fonction d'audit importante permettant de déceler les imperfections dans les habitudes des utilisateurs. Cette technique permet de détecter des familles de permissions à risques et qui nécessiterait une modification. La détermination des familles à risque peut être laissée à l'entière appréciation de l'administrateur du système, elle dépend considérablement de plusieurs facteurs tels que la fonction de l'utilisateur dans l'organisation, la valeur des informations, l'environnement organisationnel, La commande *find* et ses multiples options permet de lister les fichiers et répertoires des utilisateurs en se basant sur divers critères de sélection et fourni une aide à l'administrateur dans sa recherche des ressources mal protégées. Tout comme l'aide à la gestion de mot de passe, ce service d'audit peut se réaliser de manière manuelle mais peut également faire l'objet d'une application d'audit automatisée que nous aborderons plus loin.

Plusieurs autres centres d'intérêt doivent retenir l'attention de l'administrateur, nous en citerons quelques un dans ce paragraphe sans avoir la prétention de les avoir tous analyser mais plutôt en essayant d'attirer l'attention sur le principe de l'audit. Les fichiers cachés (dont le nom commence par un point "."), ne faisant pas partie des fichiers systèmes, présents dans les répertoires des utilisateurs sont souvent le symptôme d'une situation anormale (présence d'un cheval de Troie, ...). La détection de ces fichiers est donc un excellent moyen de déjouer certaines manœuvres. L'analyse des chemins de recherche des commandes dans les fichiers *.profile* des utilisateurs est également un excellent moyen de déceler la présence d'une situation à risque. Un autre exemple de présence anormale pourrait être celle de fichier dont le nom est identique à une commande système connue ou la présence toujours dans le fichier *.profile* des utilisateurs d'une commande d'exécution d'une de ces "fausses" commandes ou encore de copie de fichiers vers le répertoire d'un autre utilisateur. Toutes ces vérifications sont possibles au moyen des outils de base qu'offre Unix (*find*, *grep*, *ncheck*, ...) mais leur mise en oeuvre manuelle peut paraître laborieuse d'où une nouvelle fois l'intérêt de mettre au point des outils qui permettent une gestion plus facile de ces audits.

2.4.3.3. LA FORMATION DES UTILISATEURS

Le dernier aspect nécessitant l'intervention de l'administrateur du système est la formation des utilisateurs à acquérir des habitudes de travail "sécurisantes". Ce but peut être atteint par des actions diversifiées telles que la sensibilisation des utilisateurs aux problèmes de sécurité, la mise sur pied d'un cours orienté sur la sécurité du système Unix ou encore en fournissant aux utilisateurs une liste de recommandations de base leur permettant au moyen des outils fournis par Unix de se garantir un niveau de sécurité raisonnable. Notre rôle est ici de suggérer des idées aux administrateurs de système et non pas de s'immiscer dans leurs prérogatives c'est pourquoi nous développerons donc l'idée d'une liste de recommandations fournie aux utilisateurs par l'administrateur du système et leur permettant de participer à la politique de sécurisation du système telle que l'a définie l'administrateur. La liste qui suit constitue une ébauche de ce que pourrait être cette liste de recommandations.

Liste des recommandations aux utilisateurs

1. Gestion du mot de passe :

a. choix du mot de passe :

- **jamais de noms, prénoms, initiales, date de naissance**
- **jamais de noms d'objets visibles dans la pièce**
- **jamais basé sur une information personnelle**
- **au moins six caractères**
- **doit comporter des caractères spéciaux**
- **doit être aisé à retenir**
- **solutions possibles :**
 - **initiales des mots d'une phrase**
 - **mélange des lettres de deux mots non liés**
 - **suite de caractères pris au hasard**

b. utilisation du mot de passe :

- ne jamais l'écrire ou le stocker
- ne jamais le divulguer
- ne laisser personne regarder sa frappe
- ne jamais utiliser le même sur plusieurs systèmes
- différent de toute autre clé
- le changer régulièrement (sans alterner)

2. gestion des permissions d'accès :

a. aux fichiers :

- jamais en écriture pour les autres
- choix approprié des attributs
- permissions par défaut minimum (077)
- .profile accessible uniquement par soi-même

b. aux répertoires :

- jamais de répertoires en écriture pour les autres
- éventuellement -wx pour le groupe
- permissions par défaut minimum (077)

c. SUID et SGID :

- éviter les programmes SUID et SGID

3. commandes à risques :

- attention aux fichiers pouvant être copiés par cp
- attention double si ce sont des programmes SUID ou SGID
- attention au mv dans un autre système
- attention aux programmes SUID/SGID avec un nb de liens > 1
- si nb de liens > 1, faire chmod 000 avant d'effacer
- attention à l'utilisation de cpio

- jamais utiliser cpio avec l'option -u

4. chiffrement :

- chiffrer toutes les informations importantes
- choisir une bonne clé
- utiliser éventuellement makekey
- ne jamais donner la clé en argument de crypt
- utiliser l'option -x des éditeurs (sauf si makekey)
- compresser les fichiers importants avant chiffrement

5. points faibles :

a. fichier .profile :

- toujours inaccessible sauf pour soi
- répertoire le contenant jamais -wx
- surveiller régulièrement son contenu

b. fichier .exrc :

- idem .profile

c. fichiers temporaires :

- jamais dans /tmp ou /usr/tmp
- effacer les fichiers temporaires de tri dans /tmp

d. transfert de fichiers :

- chiffrer avant de transférer
- effacer les fichiers dans ... /uucppublic

e. cheval de Troie :

- surveiller le path dans .profile

- se méfier des demandes anormales de mot de passe
- se méfier des répétitions anormales de commandes
- respecter les recommandations du .profile et .exrc
- respecter les recommandations sur les accès

f. messagerie :

- filtrer les messages (attention aux !)
 - lire les messages avec le répertoire courant
- dr-x-----**

2.4.4. LES OUTILS LOGICIELS

Plusieurs outils logiciels assurant les services évoqués pour l'administrateur et même pour l'utilisateur, sont disponibles en code source C dans la littérature [WOO, 51] [SAG,44]. Ce type d'outil fournit une aide considérable dans la gestion de certains services. Les services les plus souvent visés sont ceux inhérents à la fonction d'administrateur, nous allons à cet effet évoquer dans leur grandes lignes plusieurs de ces outils et décrire leur intérêt dans la conduite d'une politique de sécurisation.

Le programme **secure** [WOO,51] est un outil d'audit de sécurité permettant à l'administrateur d'exécuter un audit sur un système Unix. Il exécute une série de vérifications sur le système et tente de mesurer son degré de sécurité. Ces vérifications incluent la recherche des programme SUID ou SGID, les fichiers lisibles et modifiables par tous les utilisateurs, les noms d'utilisateurs sans mot de passe, les noms d'utilisateurs partageant le même UID. Le résultat de l'exécution de cet outil est un rapport affiché à l'écran. L'intérêt de ce programme est qu'il s'adresse aussi bien aux utilisateurs qu'à l'administrateur, ce dernier ayant accès à des options particulières. Plusieurs options sont disponibles et notamment *-f système de fichier* qui oriente l'audit de sécurité uniquement sur le système de fichier spécifié ; *-m* (accessible par tous) donne une liste des problèmes de sécurité potentiels chez l'utilisateur qui l'a activée ; *-r (-w)* donne la liste de tous les fichiers accessibles en lecture (en écriture) pour tous à l'exception de certains fichiers systèmes bien connus (*/usr/spool/uucppublic* , */tmp* , */usr/tmp* , ...) ; *-u utilisateur* (accessible

uniquement par le superutilisateur) exécute un contrôle pour l'utilisateur spécifié ; etc. Il est évident que si le programme est activé par le superutilisateur l'audit est réalisé sur le système de fichier par défaut en entier alors que si il est activé par un utilisateur la portée de l'audit sera réduite à ce seul utilisateur.

Le code source C de ce programme étant disponible, l'administrateur peut à souhait personnaliser cette application à l'environnement qui le concerne. Nous n'avons pas cru utile de reprendre la totalité du code en annexe de ce mémoire vu son existence dans une référence répandue.

Le programme **perms** est un programme de vérification du propriétaire et des permissions d'accès des fichiers. Son intérêt est moins marquant à notre avis car il recoupe une partie des fonctionnalités du premier outil décrit et celles qui lui sont spécifiques auraient facilement pu être ajoutées au programme *secure*.

Le programme **pwadm** facilite la gestion de la durée de validité imposable par l'administrateur aux mots de passe des utilisateurs. Ce programme est accessible aussi bien pour les utilisateurs que pour l'administrateur, l'accès aux premiers est toutefois réduit à la consultation des informations relatives à la validité de son mot de passe alors que l'administrateur dispose de la faculté de modifier ces informations.

L'exécution de **pwadm** nécessite le nom de l'utilisateur sur lequel le programme doit agir, si cet argument n'est pas mentionné, le nom de l'utilisateur par défaut sera celui qui a exécuté **pwadm**. Plusieurs options sont disponibles, certaines étant désactivées lorsque l'application est exécutée par un utilisateur normal. L'option **-c** permet à l'administrateur de forcer le changement du mots de passe de l'utilisateur spécifié. L'option **-d** désactive le mécanisme de validation du mot de passe pour l'utilisateur mentionné. L'option **-m min** empêche l'utilisateur de changer son mot de passe avant un minimum de min semaines. L'option **-M max** requière de l'utilisateur une modification de son mot de passe au moins toutes les max semaines. L'option **-n** empêche l'utilisateur de remplacer son mot de passe et l'option **-p** imprime les informations de validité du mot passe pour l'utilisateur, c'est l'option prise par défaut lorsqu'aucune autre n'est spécifiée.

Ce programme assez simple nous semble être d'une aide efficace pour l'administrateur et plus particulièrement pour sa fonction de contrôle interne et de formation. Il lui permet de manière simple et relativement conviviale de gérer avec efficience les risques inhérents au mot de passe. Vu l'intérêt que cet outil

présente à nos yeux, nous croyons judicieux d'insérer son code source en annexe et de l'installer sur le serveur Electre.

Il existe des outils de chiffrement basés sur le DES tel que le programme **descript** qui chiffre l'entrée standard au moyen d'une clé chiffrée à l'aide de la commande **crypt** de Unix appliquée à la clé donnée en argument et fournit le résultat sur la sortie standard.

Le programme **suw** [SAG,44] analyse le fichier *su**log* contenant des informations sur toutes les transactions effectuées avec la commande *su* et imprime les noms de tous les utilisateurs qui se sont illégalement substitués à *root*. . Le code source C de ce programme est repris en annexe ? et cet outil a été installé sur le serveur Electre.

Vu le nombre d'outils existants et l'excellente aide qu'ils constituent pour l'administrateur du système, la mise au point d'une application supplémentaire nous semble surfaite, Il nous paraît par contre plus intéressant d'installer tous ces outils sur les hôtes de la configuration et d'éventuellement les améliorer en les adaptant aux besoins propres au site universitaire.

2.5. CONCLUSIONS

Nous avons, tout au long de ce chapitre, développé une approche bien particulière de la sécurité sur la configuration disponible à l'Institut d'Informatique. Cette modélisation client/serveur nous semble a posteriori être assez pragmatique et facile à appréhender. Elle permet de plus une évolution permanente du modèle tant en ajoutant des clients nouveaux que des serveurs supplémentaires. Les risques ne sont pas non plus limités à ceux évoqués dans la partie qui y a été consacrée, un changement totale de direction dans la fonctionnalité du système informatique étudié n'hypothéquerait nullement le modèle de base, son adaptation aux nouveaux besoins et aux nouveaux risques ne poserait aucun problème.

Les leçons que nous pouvons tirer de cette étude sont triples : la particularité de l'environnement a joué un rôle important dans les décisions prises, les services de sécurité fournis par le système d'exploitation Unix sont nombreux mais épars et enfin les perspectives d'amélioration existent et doivent être focalisée sur les trois autres serveurs, les utilisateurs, l'administrateur du système et les outils logiciels.

Le contexte dans lequel s'est développé cette étude a eu une influence prépondérante dans les prises de position et les choix qui en ont découlé. L'environnement de travail en Université est quasi entièrement basé sur la notion de coopération, de travail en équipe, de partage des informations alors que si ces idées sont présentes dans les autres organisations, publique ou privée, c'est toujours dans une moindre mesure et soumises à des règles limitatives strictes. La proposition et la conduite d'une politique en matière de sécurité logique des informations est donc très spécifique à cet environnement, elle se doit de maintenir un équilibre judicieux entre la notion de restriction et celles de partage et de communication, en laissant une certaine prépondérance à ces deux dernières.

La notion de sécurité pour utilisateurs de la configuration informatique de l'Institut signifie surtout une protection de l'intégrité de leurs informations beaucoup plus que leur confidentialité. Cette notion essentielle représente à notre avis une des grandes différences avec les autres organisations qui de par leur rôle industriel, économique, médical ou autre se doivent de maintenir une confidentialité maximale des informations.

Le système Unix peut être considéré comme un énorme boîte de "légos" mise à la disposition tant des utilisateurs que de l'administrateur, leur

fournissant les possibilités de gérer une grande partie des problèmes de sécurité. Le rôle des utilisateurs et de l'administrateur du système est de construire au moyen de ces "légos" un édifice solide représentant une politique cohérente en matière de sécurité. On peut considérer que le système Unix offre une sécurité de niveau C dans la hiérarchie établie par le **US DoD**, ce niveau suffit si on prend en considération la première leçon tirée au paragraphe précédent, le passage au niveau B de cette hiérarchie est possible et a été entrepris dans des organisations où la sensibilité des informations le nécessitait [CHU,11]. Cet accroissement du niveau de sécurité serait de plus en contradiction avec l'équilibre évoqué au paragraphe précédent. Le système tel qu'il est conçu nous paraît donc suffire à une bonne gestion de la problématique de la sécurité au sein de la configuration de l'Institut d'Informatique.

La suffisance de l'environnement de base ne signifie nullement qu'elle représente un aboutissement dans ce domaine, il existe en effet des perspectives d'améliorations intéressantes. Ces perspectives sont toutefois à placer dans un contexte plus managérial que technique. Les améliorations doivent en effet naître dans le chef des utilisateurs et de l'administrateur du système. Les utilisateurs se doivent de prendre conscience des risques mais également de l'utilité de leur participation dans la politique de sécurité. Il est important que ces acteurs essentiels n'éludent pas leurs responsabilités dans un domaine que l'on peut comparer à une chaîne dont la résistance est dépendante de celle du maillon le plus faible. Si l'on veut maintenir l'analogie avec le monde du cinéma, on pourrait dire que l'administrateur du système est le metteur en scène. Il se doit de contrôler et de coordonner la participation des utilisateurs, les renseigner voir les former à utiliser des méthodes sécurisantes, garantir la réalisation de leurs souhaits. Une sorte de consensus de travail doit donc être établi entre l'administrateurs et les utilisateurs, garantissant l'équilibre nécessaire au bon fonctionnement du système et de l'organisation. Les outils logiciels constituent également un domaine prometteur en matière de sécurité, et particulièrement comme outil d'aide à la gestion des ressources de sécurité. Il nous paraît important que tant l'administrateur que les utilisateurs en fassent un usage important et efficace. Leur implémentation ne devrait guère poser de problème, seule l'adaptation aux besoins spécifiques et l'installation de ces outils existants constitue le travail technique préalable à leur utilisation.

Au terme de cette étude du problème de la sécurité dans un environnement informatique spécifique, nous pouvons considérer que la

sécurité est avant tout un problème de politique et de choix. La tendance actuellement étant de surprotéger, la première question qu'il nous paraît pertinent de se poser concerne la nécessité d'une sécurité informatique élevée, et ce sur base des besoins ressentis par les différents acteurs et les risques auxquels ils sont soumis. Cette analyse initiale nous semble être la garante d'une efficacité de la politique choisie et par là-même de la satisfaction des utilisateurs et de l'augmentation de leur productivité.

CHAPITRE 3 :

LES PERFORMANCES

3.1. INTRODUCTION

Les études de performance de systèmes informatiques peuvent être classifiées suivant leur objectif. Les besoins de beaucoup d'installations trouvent leur réponse dans l'un des types d'étude de performance suivants : les études de sélection, d'amélioration, de planification, de capacité ou de conception.

Les études de sélection ont pour but d'identifier le système qui répond le mieux à un ensemble de critères de sélection.

Les études d'amélioration concernent les modifications apportées à un système existant pour améliorer ses performances ou ses fonctionnalités, ou encore pour diminuer son coût.

Les études de planification de capacité sont utilisées pour prévoir jusqu'à quel point le système pourra répondre au niveau de performance désiré, si la charge de travail augmente.

Dans tous ces cas d'études, l'analyste doit disposer d'informations pour répondre aux questions concernant les performances du système. Deux types d'informations peuvent être utilisées en mesure de performance : des données réelles ou une modélisation. Bien qu'une modélisation soit totalement en dehors de nos objectifs, il est utile de signaler qu'elle peut constituer un outil utile dans l'arsenal de l'analyste des performances et être complémentaire des mesures directes de performance.

Dans tout système informatique, l'utilisation effective de mesures est essentielle pour le diagnostic et l'anticipation des problèmes. L'identification des goulots d'étranglement, par exemple, est possible seulement si des mesures sont effectuées avec un large éventail de charges de travail, permettant ainsi d'observer les causes et la localisation des engorgements. L'analyse des données mesurées peut aussi indiquer quels paramètres du système peuvent être modifiés pour diminuer les points de congestion.

Si l'utilité d'un moniteur de prise de mesures de performance n'est maintenant plus à démontrer, il reste encore à choisir le type d'outil de mesure qui sera le plus adapté à notre problème. Sans perdre de vue les objectifs que nous nous sommes fixés, il nous faudra également déterminer quelles seront les mesures que le moniteur collectera. Il s'agira là d'évaluer l'utilité et la pertinence , pour le gestionnaire du système, des différentes grandeurs caractérisant un système informatique. Un élément dont l'importance est généralement sous-estimée dans l'évaluation des performances, est la présentation et la lisibilité des résultats. On veillera donc également à ce que notre outil réponde à toutes les attentes de son utilisateur en ce qui concerne ce critère.

3.2. ANALYSE DU PROBLEME

3.2.1. EVALUATION DES BESOINS

3.2.1.1. LES BESOINS GLOBAUX

Sélectionner les mesures de performance pour un système en temps réel équivaut à définir une série d'aspects des performances du système, qui ont un effet sur la productivité de l'utilisateur. Les mesures de performance les plus largement utilisées pour évaluer un système sont :

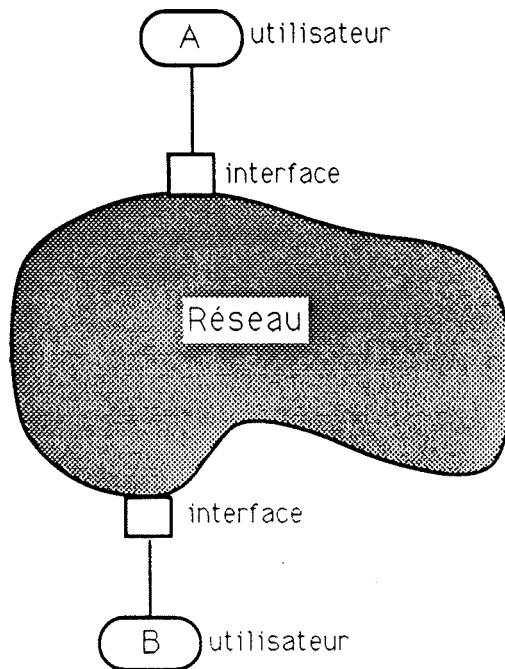
- le temps de réponse au terminal
- le pourcentage de disponibilité du système
- le volume des transactions pour une période donnée
- la charge des lignes de communication
- le délai moyen de transfert
- le pourcentage de travaux exécutés à temps
- le pourcentage de réexécution
- l'utilisation des machines

3.2.1.1.1. Le temps de réponse

Le temps de réponse est sans conteste la notion de performance la plus perceptible pour un utilisateur. Une étude récente de A. Rushinek et S. Rushinek [RUS, 42] a montré qu'un bon temps de réponse du système est un facteur majeur contribuant à la satisfaction de l'utilisateur. D'autres recherches ont montré que la productivité de l'utilisateur augmentait pendant les périodes durant lesquelles le temps de réponse était meilleur [BAR , 4]. Celui-ci a même un impact, selon une étude récente [GUY, 25] , sur l'état d'anxiété de l'utilisateur.

Le temps de réponse est défini comme le temps requis pour l'exécution d'une transaction, depuis la transmission du message à partir du terminal, jusqu'à la réception du résultat de la transaction par ce même terminal [AUE,2]. D'un point de vue plus orienté réseau de communication, on peut considérer le temps de réponse comme le temps pour transmettre correctement un message d'un utilisateur

A vers un utilisateur B, en ce compris la décomposition en paquets, et l'encapsulage et de recevoir la réponse [HAM, 26].



Le temps de réponse se décompose en fait sur un réseau en plusieurs parties [HAM ,26] :

1. Le temps pour les paquets du message de l'utilisateur A de traverser la connexion vers l'interface de la station.
2. Le temps pendant lequel les paquets doivent attendre avant d'être transmis.
3. Le temps nécessaire à la transmission des paquets composant le message à travers le réseau, en ce compris le temps de propagation.
4. Le temps d'attente des paquets dans l'interface de l'utilisateur B.
5. Le temps requis pour traverser la connexion de l'utilisateur B.
6. Le temps pour l'utilisateur B de générer une réponse.
7. Le temps pour la réponse de revenir à travers le système suivant les étapes 1 à 5.

La transmission d'un message par un utilisateur à son terminal est généralement signalée par la pression d'une touche particulière au clavier. La réception du résultat de la transaction varie d'un système à l'autre. Cela peut être le résultat définitif du message initialement transmis, mais cela peut également être un message signalant ou suggérant le passage à la transaction suivante. Dans tous les cas, le temps de réponse au terminal sera le temps durant lequel l'utilisateur devra attendre avant de commencer une transaction après avoir terminé la précédente.

Le problème du temps de réponse est sa dépendance envers de multiples facteurs caractérisant le contexte de travail.

Le type d'application a un impact important, non seulement sur le temps de réponse en lui-même, mais aussi sur sa perception par l'utilisateur. Il est évident qu'une transaction demandant de nombreux accès disque occasionnera un temps de réponse beaucoup plus long qu'une autre ne nécessitant qu'un rapide calcul. L'impact du type d'application se fait également sentir sur la perception qu'a l'utilisateur du temps de réponse. Une application ne nécessitant que peu de réflexion de la part de l'utilisateur aura pour conséquence de lui faire percevoir le temps d'attente comme beaucoup plus long que dans le cas d'une application demandant une réflexion en attendant le résultat. C'est typiquement le cas d'une application d'introduction de données.

La charge du réseau est un facteur de variation important du temps de réponse, si l'exécution de la transaction nécessite un transfert d'informations via ce réseau. Cet impact peut être considéré comme peu important, tant que la capacité maximale n'est pas approchée.

Si on considère le système comme étant multiutilisateur et multitâche, son taux d'utilisation au moment de la mesure du temps de réponse a un impact important sur ce dernier.

Dans le cas d'un poste de travail plus évolué qu'un terminal, telle une station de travail, il faut également tenir compte de la charge du processeur de cette station.

Pour avoir une applicabilité plus générale, la plupart des mesures de performance sont exprimées en terme de moyenne,

plutôt qu'en terme de variables dépendant de l'état instantané du système au moment de la mesure. La nécessité d'utiliser une telle technique est évidente pour le temps de réponse, vu les nombreux facteurs de variation évoqués. Plusieurs grandeurs statistiques de la mesure du temps de réponse sont intéressantes, notamment l'écart-type ou encore la variance.

3.2.1.1.2. Le pourcentage de disponibilité

Le pourcentage de disponibilité du système est en relation étroite avec le temps de réponse au terminal. Plus le pourcentage de disponibilité approche zéro, plus le temps de réponse approche l'infini. Un mauvais pourcentage de disponibilité, et par la même un piètre temps de réponse, peuvent être les symptômes de différents problèmes.

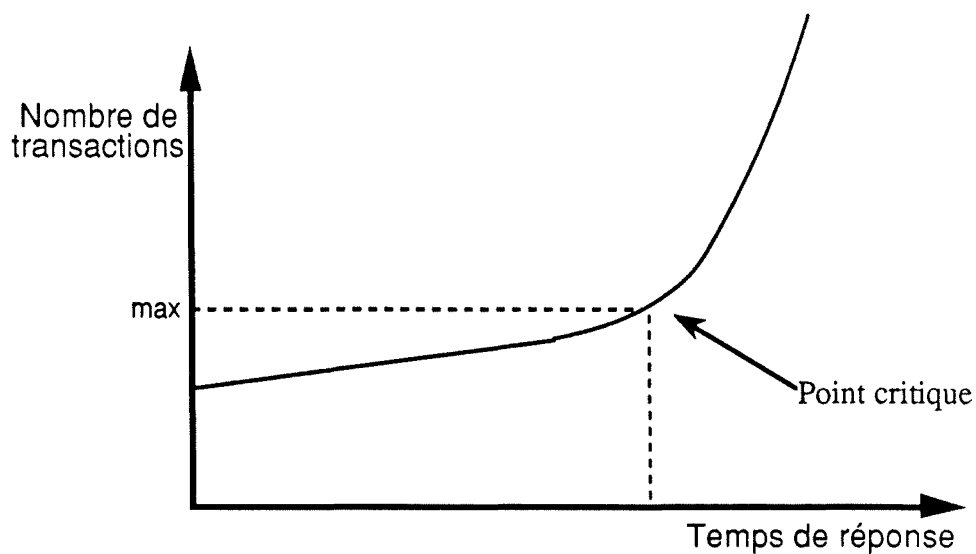
Le pourcentage de disponibilité est une des mesures de performance préférées, vu sa facilité de calcul. Il est déterminé en utilisant la formule suivante : temps total - temps de panne

temps total

où le temps total est considéré comme le nombre d'heures durant lesquelles le système est en ordre de fonctionnement pour une période donnée. La période peut être un jour, une semaine, un mois ou un an. Le temps de panne est le nombre total d'heures durant lesquelles le système est indisponible pour la même période.

3.2.1.1.3. Le volume de transactions

Le volume de transactions est typiquement le nombre de transactions exécutées durant une heure. L'acceptation de la transaction est fonction de l'organisation et du type d'application. Chaque système a un point critique à partir duquel le temps de réponse se dégrade rapidement. Ce point est généralement mesuré en terme d'un nombre maximum de transactions par heure, et est déterminé durant les tests d'initialisation du système.



L'évolution typique du temps de réponse, par rapport au nombre de transactions, suit une courbe à tendance exponentielle similaire à celle de la figure précédente [AUE, 2].

3.2.1.1.4. La charge des lignes de communication

Le volume d'informations circulant sur une ligne de communication n'est pas seulement une mesure de performance, mais également un facteur d'évaluation de l'efficacité d'un réseau de communication. L'analyse de la charge des lignes peut révéler le besoin de changer la configuration du réseau et, par ce fait, isoler les problèmes de performance pouvant être solutionnés ou prévenus.

La charge, exprimée en bits par seconde, peut être définie comme le nombre moyen de bits passant à un point donné sur un réseau par unité de temps [HAM,26]. Une seconde approche de la charge, plus restrictive que la précédente, consiste à ne prendre en compte que les bits composant les paquets corrects, il s'agit alors d'une charge efficace. La charge peut être appliquée au réseau entier ou à une partie, dans ce dernier cas la charge est le nombre moyen de bits par seconde entrant et sortant de la section du réseau.

La charge est fréquemment normalisée en la divisant par la capacité de transmission du support, de manière à obtenir un nombre situé entre zéro et un. La charge normalisée peut être obtenue par la formule :

$$S = k * X / R$$

où S est la charge normalisée , k le taux d'entrée moyen à la section du réseau étudiée et exprimé en paquets par seconde, X est le nombre moyen de bits par paquet et R est la capacité de transmission du support.

La charge peut être appliquée au réseau complet, à une section du réseau mais aussi à une simple connexion [HAM,26]. Quand elle est appliquée au réseau complet, la charge non normalisée représente le nombre total de bits par seconde circulant sur le réseau. Les mesures faites sur les différentes connexions peuvent être sommées pour obtenir la charge totale du réseau.

La charge normalisée est étroitement liée à une autre mesure, appelée l'utilisation ou l'efficacité du canal, et définie comme la fraction moyenne de temps durant laquelle le canal est occupé [HAM, 26]. Si le support transporte seulement des paquets corrects quand il est occupé et qu'on ne tient pas compte de l' **overhead**, alors la charge efficace normalisée et l'utilisation sont égales. L'utilisation du canal peut être utilisée dans un sens plus spécifique, elle peut alors être définie comme la fraction de temps durant laquelle le canal transmet des données, par opposition à la transmission des données et des overheads. Si D est le nombre de bits de données dans un paquet et H le nombre de bits d' overhead dans ce paquet, l'utilisation du canal est $S * D / (D + H)$.

3.2.1.1.5. Le délai moyen de transfert

Le délai de transfert est défini comme le temps depuis l'arrivée du dernier bit d'un paquet dans une station du réseau jusqu'à ce que le dernier bit de ce paquet soit délivré à travers le réseau à la station

de destination. Comme pour la charge, il est possible de normaliser le délai de transfert moyen T , en utilisant les variables définies dans le point précédent, par la formule suivante :

$$T' = T / (X / R) = RT / X$$

sachant que X / R secondes par paquet est le facteur de normalisation.

3.2.1.1.6. Le pourcentage de travaux exécutés à temps

Une des premières mesures à prendre en compte dans un système à exécution différée est le pourcentage de travaux exécutés à temps [KIN,29]. Le gestionnaire du système devrait avoir une liste disponible et à jour des travaux qui n'ont pu se réaliser correctement, celle-ci devant reprendre une explication pour chaque délai. Un résumé hebdomadaire ou mensuel des échecs, par travail et par cause, peut être d'une aide importante dans l'identification des zones à problème.

3.2.1.1.7. Le pourcentage de réexécution

Une des causes principales du non respect des délais d'exécution des travaux est la nécessité de les réexécuter. Le temps de réexécution peut être défini comme la quantité de temps requise pour l'exécution d'un travail qui a échoué [KIN,29]. Réexécuter un travail retarde non seulement ce travail, mais affecte également l'ordonnancement des autres. Faire attention au temps de réexécution peut être d'une grande importance dans l'identification des problèmes. Si des détails supplémentaires sont désirés, un rapport de réexécution peut être préparé pour chaque travail important exécuté. Pour avoir une vue évolutive du problème, le pourcentage de temps de réexécution par rapport au temps total d'exécution peut être présenté graphiquement.

3.2.1.1.8. L'utilisation des machines

Une composante du rôle de l'administrateur du système est d'assurer une utilisation efficiente des ressources. Mesurer la réussite de cet objectif implique un suivi de l'utilisation des machines. Les aspects pouvant être mesurés sont nombreux et variés. Ils recouvrent l'utilisation du ou des processeurs, de la mémoire, des mémoires de masse, des imprimantes ou de tout autre périphérique.

3.2.1.2. LES BESOINS SPECIFIQUES

Les différents grandeurs pouvant caractériser un système informatique ont été présentées au point précédent. La sélection, parmi ces grandeurs, des informations qui répondent à l'attente du gestionnaire de la configuration est une étape difficile.

L'avis des personnes directement concernées nous est paru indispensable; c'est pourquoi nous avons demandé, aux responsables des différent éléments constituant le réseau, quelles étaient les mesures dont ils aimeraient disposer.

De ces interviews, il ressort que cinq domaines principaux de mesures retiennent l'attention de ces experts : l'occupation des CPU, les accès aux disques, l'utilisation des mémoires, l'activité de pagination et la charge du réseau.

L'occupation des processeurs des différentes machines de la configuration peut être caractérisée par différentes mesures.

L'équilibre du système est mesuré par le pourcentage d'utilisation du CPU dans les différents états qu'il peut prendre, à savoir les modes système, utilisateur et l'état endormi. Un système équilibré passe la plupart de son temps en mode utilisateur (40-60%), quelques temps dans l'état endormi et le reste en mode système.

Si l'équilibre du système est bon, mais que l'on constate un ralentissement des machines, il faut alors porter son attention sur la charge moyenne. La charge moyenne (**load average**) est une grandeur classique de caractérisation du CPU. Elle mesure la longueur moyenne de la file d'attente du processeur. La charge moyenne varie suivant la rapidité et l'utilisation du processeur (utilisation orientée calcul ou entrée/sortie, ...).

Une autre mesure indispensable, la distribution des travaux, reprend le nombre de processus en exécution, bloqués et en attente. Cette mesure aussi peut aider à se faire une idée de l'état du système. Un nombre important de processus bloqués peut, par exemple, être le signe d'une surcharge ou d'un mauvais équilibre du système.

Il est souvent possible de mettre le doigt sur la cause d'un temps système élevé en regardant s'il y a un nombre de changements de contexte, d'interruptions ou d'appels systèmes excessif. Si, par exemple, le nombre d'interruptions est élevé, il faut chercher un éventuel appareil très occupé.

Deux autres domaines requièrent impérativement des mesures : les accès aux disques et l'utilisation des mémoires. Des informations concernant les partitions spécifiques des disques, incluant l'espace occupé par les principaux systèmes de fichiers, l'espace utilisé et disponible, doivent être fournies.

L'activité de pagination constitue également un bon témoin de l'état du système. Si la mémoire virtuelle est activée, le **démon** de pagination s'exécutera. Il est déclenché dès que la quantité de mémoire libre descend en dessous d'un certain seuil et il augmente son activité lorsqu'elle tend vers zéro. Ainsi, si le système est lourdement chargé ou s'il y a peu de mémoire disponible, le système sera forcé de paginer. Ceci est généralement accompagné d'une diminution non négligeable des performances du système et de longues pauses lorsque des travaux interactifs, tels des éditeurs ou des programmes de fenêtrage, paginent vers la mémoire. On sera donc particulièrement vigilant vis-à-vis de ces données si l'on s'attend à travailler dans un environnement pauvre en mémoire pour une longue période, ceci pouvant entraîner une limitation de la charge de travail.

Un cinquième domaine mérite aussi toute notre attention : la charge du réseau.

Etant donné le nombre relativement réduit de machines utilisant effectivement le câble du réseau, on peut écarter tout risque de dépassement de capacité de ce câble. On s'attachera donc plutôt à mesurer la contribution au trafic du réseau de chaque machine connectée, c'est-à-dire combien de paquets sont transmis et combien sont reçus . On pourra, par exemple, constater que le nombre de paquets entrant vers un serveur augmente chaque fois qu'un utilisateur se connecte sur un hôte, alors que le nombre de paquets sortant reste invariable. Ceci suggère que le serveur voit les paquets de demande de connexion de l'hôte, mais ne réalise pas qu'il est supposé lui répondre. Ceci peut être dû à une adresse incorrecte (dans le fichier /etc/hosts en Unix). Si, dans le cas contraire, le nombre de paquets entrant reste lui aussi stable, la machine ne voit alors aucun paquet. On peut envisager, dans ce cas, un type de problème différent, probablement un problème matériel.

Le nombre d'erreurs et de collisions devra aussi être pris en compte. Un taux d'abandon des paquets de plus de trois pourcents fera suspecter un problème matériel à une des interfaces du réseau, on contrôlera aussi les transceivers et le câble Ethernet.

Les données indispensables à une bonne gestion du système sont maintenant cernées. Une liste des mesures qui seront à collecter par le moniteur sera dressée dans le point traitant des choix que nous avons effectués.

3.2.2. ANALYSE DES OUTILS DE PRISE DE MESURES

3.2.2.1. OUTILS GENERALEMENT UTILISES

3.2.2.1.1. Les outils de prise de mesures

Traditionnellement, les outils de mesure sont classifiés en outils matériels et outils logiciels. Ces classifications ne sont pas en rapport avec la nature des paramètres à mesurer; en principe, les performances d'un programme peuvent être mesurées par un outil matériel et les performances d'un matériel par un outil logiciel [FER, 18]. Les outils matériels sont constitués d'un système matériel ajouté à la configuration dans un but de prise de mesures. Les outils logiciels sont composés d'instructions à être exécutées par le système de mesure.

(a) Les moniteurs matériels

(1) Description

Le suivi du fonctionnement détaillé des composants d'un système informatique et l'identification des traitements qu'il effectuent, sont possibles grâce au relevé des divers signaux électroniques qui régissent toutes les activités d'un ordinateur. D'où l'idée de prélever l'état des signaux intéressants par un appareil externe muni d'un certain nombre de sondes actives à haute indépendance, branchées à différents points de capture sur le système informatique étudié.

Ces appareils sont composés d'organes de saisie, d'une unité de traitement et de dispositifs de stockage. Le dépouillement des mesures était à l'origine effectué sur l'ordinateur mesuré (ou sur un autre) à l'aide d'un logiciel fourni par le constructeur. La plupart des moniteurs matériels comporte maintenant d'une façon plus ou moins intégrée, un mini-ordinateur permettant d'effectuer localement ou à distance le dépouillement, et de gérer une base de données des mesures effectuées. Il existe également diverses possibilités de

visualisation en temps réel des mesures et de pilotage automatique des campagnes [SAB, 43].

(2) Avantages et inconvénients

Comme ces moniteurs ont généralement leur propres ressources, ils occasionnent peu ou pas d'interférence avec le système à mesurer. Des interférences peuvent cependant se produire si les données collectées sont périodiquement stockées sur le système mesuré ou s'il est permis au moniteur d'accéder à la mémoire principale du système mesuré [FER, 18].

Si ce n'est pour les temps de réponse au terminal pour des lignes et des terminaux individuels, le moniteur matériel n'est pas équipé pour mesurer les performances d'applications. De plus, ces outils peuvent être coûteux et exigent généralement une connaissance du système à mesurer digne d'un expert.

(b) Les moniteurs logiciels

(1) Description

Un moniteur logiciel est un ensemble de programmes spécialisés, externe au système d'exploitation, assurant à la fois les fonctions d'extraction des données concernant l'utilisation d'un système informatique et d'analyse de ces informations. La différence avec les moniteurs matériels qui remplissent les mêmes fonctions, consiste dans la saisie par module logiciel des valeurs significatives et par le chargement dans le système mesuré. Les moniteurs logiciels ne sont pas universels et ils doivent comporter une version propre à chaque type d'exploitation. Ils ne sont donc pas intégrés directement au système d'exploitation.

Ces outils sont, d'autre part, plus orientés vers les mesures relatives au fonctionnement interne des éléments de la configuration que vers l'affectation à chaque travail des

ressources consommées , bien qu'ils puissent parfois accomplir cette tâche. Ils comportent éventuellement des moyens propres d'analyse des mesures effectuées [SAB, 43].

(2) Qualités requises

Le moniteur logiciel doit pouvoir accéder au maximum de données reflétant le fonctionnement du système informatique pendant son utilisation. Ceci s'effectue par la consultation de compteurs et de tables en mémoire centrale, ainsi que de certaines données utilisateurs. Son activation doit être facile et rapide. De plus, de même qu'il est possible de déplacer les sondes d'un moniteur matériel, l'introduction de nouvelles mesures doit pouvoir s'effectuer sans modifications trop importantes.

Bien que ne faisant pas directement partie du système d'exploitation, le moniteur est inséré dans ce système. Son installation pouvant nécessiter quelques aménagements, les conséquences de celles-ci doivent être entièrement transparentes lorsque l'outil de mesure n'est pas activé. Il est par contre impossible d'éviter totalement une perturbation durant son fonctionnement, celui-ci nécessitant certaines ressources : temps unité centrale, mémoire et dispositif de stockage. On cherchera à limiter au maximum les conséquences de cette perturbation et à déterminer dans les mesures effectuées, la part générée par le moniteur lui-même afin de corriger au besoin les défauts.

(3) Fonctionnement

La première phase est la phase d'extraction. La saisie de l'information est réalisée par des modules logiciels comportant des séquences d'instructions pour la lecture des compteurs, tables et données. Chacun de ces modules peut prélever une ou plusieurs informations. Le nombre de modules n'est pas limité si ce n'est par l'overhead qu'ils induisent. Pour faciliter la diversité des mesures et leur évolution, il est préférable d'adopter une structure très diversifiée, chaque module

collecteur étant affecté, par exemple, à la saisie d'un seul type d'information.

La prise de mesure s'effectue de manière discrète dans le temps. Son déclenchement est provoqué, soit par un événement système (**event-driven**), soit à périodicité constante (**sampling**). La première méthode permet le relevé exhaustif des événements mesurés, la seconde apporte un échantillon qui est la plupart du temps suffisant pour la détermination des valeurs. Certains moniteurs sont constitués d'une combinaison de ces deux modes de prélèvement. Lors du fonctionnement en mode temps, on parle d'une "photographie" toutes les n secondes, n pouvant être supérieur ou inférieur à 1. Cette forme d'extraction nécessite au préalable une description précise des ressources à mesurer, paramétrables en fonction des diverses configurations.

Lors de la phase de pré-traitement et de stockage, les informations sont tout d'abord stockées en mémoire dans des compteurs ou des tables. Au besoin, certains traitements intermédiaires sont effectués avant de stocker les données sur disque. Afin de réduire l'interférence entre les phénomènes mesurés et l'outil de mesure, on reporte à la phase de dépouillement le maximum de traitements. L'exploitation doit surveiller qu'en cas de reprise après incident-machine ou d'autres arrêts, le moniteur logiciel est bien réactivé, sous peine de perdre des périodes entières d'observations. De même, il convient en fin de session d'activer un module spécifique de clôture de fichiers.

La phase de dépouillement s'effectue en général en différé. Bien qu'envisageable, l'affichage en temps réel est peu développé dans les outils actuellement utilisés. La plupart de ces logiciels permettent de publier les états à différents niveaux de détails, l'une des principales difficultés de l'interprétation des mesures étant de savoir, en fonction des besoins, jusqu'où aller dans les investigations. En effet, beaucoup de moniteurs logiciels sont utilisés de manière permanente et la publication

régulière d'un trop grand nombre de données n'assure pas une bonne utilisation de celles-ci.

(4) Avantages et inconvénients

Puisque le moniteur logiciel est composé d'instructions additionnelles qui doivent être exécutées par le système à mesurer, une certaine interférence est inévitable. De plus, comme il n'est qu'un programme d'application, il ne peut obtenir les services du CPU, lorsque des services de haute priorité sont en cours de réalisation. La précision des outils logiciels est donc généralement plus faible que celle des outils matériels.

Cependant, contrairement aux moniteurs matériels, les moniteurs logiciels peuvent être utilisés pour mesurer les performances de programmes d'application, et pas uniquement pour la saisie de grandeurs sur l'utilisation du matériel. Si des ressources sont retenues sans être utilisées, la nature de l'attente peut être examinée. Le moniteur logiciel peut effectivement fournir de telles informations en renseignant, par exemple, les travaux en mémoire en attente du CPU. En multiprogrammation, le temps utilisé par un programme pour accomplir une opération n'est pas vraiment significatif. L'important est de savoir quelle autre activité est présente lorsqu'un programme particulier s'exécute. Le moniteur logiciel peut alors aider à identifier les travaux incompatibles qui causent des goulots d'étranglement [AUE,2].

Le moniteur logiciel est bien souvent beaucoup plus souple que les outils matériels. Il est plus facile à installer, bien qu'une certaine expérience est généralement nécessaire pour son installation, son utilisation et l'interprétation des résultats.

3.2.2.1.2. Les outils de présentation

Certains outils de prise de mesures sont équipés d'un module de réduction des données et de génération de rapports qui calcule des statistiques et les présente sous forme de tableau ou de graphique. Cependant, il est difficile d'interpréter

des résultats quand de nombreuses grandeurs sont à considérer simultanément. Une présentation efficace peut constituer un sérieux bénéfice, et, peut être plus grave, une mauvaise présentation peut cacher des résultats importants. Ceci est particulièrement vrai pour des résultats qui ne sont pas attendus par le gestionnaire, et qui sont souvent les plus intéressants.

Pour l'analyse des résultats, la possibilité d'une vision quasi brute des données peut être une alternative intéressante aux inférences statistiques rigoureuses. Elle constitue une approche directe qui conduit souvent à des premières conclusions pertinentes qui peuvent, par la suite, être examinées plus rigoureusement [FER, 18]. Cette technique de présentation doit procurer aisance, rapidité et facilité d'interprétation. Elle doit inclure :

- un réarrangement des données
- un calcul de statistiques simples (moyenne, variance, ...)
- une construction de graphiques simples qui permet à l'oeil humain d'être utilisé comme détecteur de caractéristiques

Dans tous les cas, les résultats des tests sont présentés simplement sous forme de tableaux. Bien que ces tableaux sont faciles à produire, leur limitation bidimensionnelle les rendent inappropriées quand plus de deux quantités varient simultanément. L'existence de relations, de caractéristiques ou de tendances importantes peuvent donc ne pas être remarquées.

Les représentations graphiques peuvent conduire à la découverte de relations plus compliquées. Par exemple, la dépendance entre deux facteurs différents peut être mise en évidence par la présentation en courbe. Un module de présentation d'utilisation aisée est donc indispensable pour permettre la production de tels graphiques.

3.2.2.2. OUTILS DISPONIBLES

Deux types d'outils peuvent être distingués au sein de la configurations des hôtes sur le réseau telle qu'elle a été décrite au chapitre 1. Les outils de base du système d'exploitation sont généralement des commandes renvoyant une série de mesures relativement éparses. Deux outils de plus haut niveau existent sur les machines Unix : *DxTopCPU* et *Monitor.t30*. Nous allons aborder en détails ces derniers, alors que nous nous contenterons de vous

renvoyer à l'annexe D pour plus d'informations sur les résultats accessibles au moyen des commandes de base.

3.2.2.1.1. DxTopCPU

(a) Description

DxTopCPU est un programme qui peut travailler en deux modes. Le premier mode permet la surveillance du processeur et le second l'affichage graphique d'informations concernant la charge de la machine. Les données renseignées par le premier mode sont par processus :

Données	HISTOG.	VALEUR
le pourcentage d'utilisation du CPU	X	X
le nom de la commande		X
la taille résidante	X	X
le temps CPU total	X	X
le nom de l'utilisateur		X
l'identificateur du processus		X
la priorité du processus	X	X
la taille totale	X	X
l'état courant		X
le UID		X

Un X dans la colonne HISTOG. signifie que le programme peut afficher cette donnée en histogramme, et un X dans la colonne VALEUR indique que cette mesure peut être présentée en valeur numérique.

Dans le second mode, les informations suivantes peuvent être tracées sous forme d'une courbe évolutive :

La charge totale du CPU en pourcent (100% - endormi)

La charge du CPU en mode système en pourcent

La quantité de mémoire libre

Le nombre total de transferts du disque par seconde

Le nombre total d'interruptions par seconde

Le nombre total de paquets entrant et sortant ainsi que les collisions

Le programme peut aussi affiché, en mode graphique, une entête reprenant un résumé des informations suivantes :

Date

Heure

Nom de l'hôte

Nombre de processus et leur état

Pourcentage de temps dans chaque état

Etat de la mémoire (réelle/virtuelle/libre)

L'application utilise un menu de type pop-up pour permettre à l'utilisateur de quitter, geler ou dégeler le programme et modifier les paramètres. La commande paramètre donne accès à une boîte de dialogue qui permet de sélectionner un des deux modes d'affichage décrit ci-dessus. Le mode par défaut au lancement étant le mode tableau. Il permet également de demander ou non l'affichage de l'entête et de choisir l'intervalle de prise de mesure en secondes, ainsi que les données que l'on veut voir apparaître dans le mode graphique.

La caractéristique principale de cet outil est d'offrir à l'utilisateur une prise continue de mesures lui affichant immédiatement les résultats à l'écran. Sa durée de vie est donc entièrement limitée à la durée de la session de l'utilisateur.

(b) Avantages

Le premier avantage de ce type d'outil découle logiquement de son principe, il permet à tout utilisateur de surveiller en permanence l'état de la machine sur laquelle il travaille. Généralement les utilisateurs affichent les résultats de ce type d'outil en arrière-plan de leur tâche principale même si, dans beaucoup de cas, il sert plus à la décoration de l'écran qu'à une réelle surveillance du système.

Ce logiciel pourrait être utilisé comme un mesureur pseudo event-driven, en ce sens que le déclenchement doit être réalisé par l'utilisateur lorsqu'un événement particulier se produit tel que surcharge, panne locale, etc.

La variété des mesures disponibles est relativement grande et correspond assez bien aux besoins exprimés tant par les gestionnaires de système que par les utilisateurs avertis ou plus intéressés par le domaine des performances.

L'affichage graphique offert par DxTopCPU lui donne un attrait supplémentaire. Ce type d'affichage permet, même en arrière-plan, d'attirer l'attention de l'utilisateur de façon beaucoup plus évidente qu'un tableau de valeurs.

(c) Inconvénients

Le manque totale de documentation tant au niveau de l'utilisation que du point de vue des définitions précises des mesures prises se fait cruellement sentir. Si l'utilisation ne pose que peu de problème au départ, grâce à la convivialité du logiciel, l'absence de définitions précises des mesures effectuées risque de poser un problème évident d'interprétation des résultats.

Le graphisme tel qu'il est utilisé dans cet outil semble moins utile qu'agréable à voir. Cette technique de graphique évolutif, dont les courbes défilent sous les yeux généralement béas d'admiration des utilisateurs, a un caractère pratique très réduit pour deux excellentes raisons. Les résultats ne sont pas sauvés, donc dès qu'ils disparaissent de l'écran ils sont perdus pour l'utilisateur. De plus l'absence d'échelle précise cumulée à la taille réduite, voir très réduite, du graphique, en fonction de l'espace écran nécessaire pour les autres applications, fait qu'une lecture correcte est impossible. Son seul intérêt réside donc dans sa capacité à attirer l'attention de l'utilisateur ou de l'administrateur lorsqu'une variation importante survient.

L'absence de possibilité de faire fonctionner cet outil en toile de fond comme un démon système élimine la possibilité de l'utiliser comme outil d'analyse. C'est réellement un outil de surveillance et rien d'autre.

3.2.2.1.2. Monitor

(a) Description

Monitor est un utilitaire de surveillance des performance d'un système. Il permet de collecter une série de données sur un système en fonctionnement et affiche les informations à l'écran ou les sauvegarde dans un fichier. Ce fichier peut être réutilisé par un programme utilisateur ou repris par Monitor.

Le programme et le format d'affichage ont été inspirés du programme "mon" de Philippe DYKSTRA. La fonctionnalité de sauvegarde et récupération a été, quant à elle, inspirée de l'utilitaire Monitor sur VMS.

Les données mesurées par cette application et supportées, tant par le mode affichage que par le mode sauvegarde, sont :

DESCRIPTION	DONNEES
Utilisation du CPU, interruptions, ...	cpu
Collection de données par processus	pid
Compteur d'accès au terminal	tty
Données concernant les E/S disque	disk
Mesure de la mémoire libre en pages	free
Compteur de forks et vforks	fork
Nombre de processus en différents états	processes
Information de pagination	page
Nombre d'utilisateurs dans le système	users
Données sur l'interface réseau	netif
Utilisation de la mémoire	memory
La "charge moyenne"	Loadave

Il existe également, en plus de l'accès par domaine qui vient d'être évoqué, un accès par groupe de données :

DESCRIPTION	GROUPES
Disk + netif + tty	io
cpu + fork + page + memory	os
Toutes les données	mon

Le logiciel s'exécute grâce à une ligne de commande de type Unix classique :

monitor [mode] [mode d'affichage] [options] données

Le mode est la méthode utilisée pour la collection des données. Trois modes différents sont accessibles "live", "names" et "replay". En mode "live" les données sont collectées directement sur le système en fonctionnement. En mode "replay" les données proviennent d'un fichier précédemment créé par une exécution de Monitor. Enfin en mode "names" Monitor imprime un listing de la configuration, cette option n'a pu être vérifiée. Le mode par défaut étant le mode "live".

Les mode d'affichage accessibles pour sortir les données collectées sont "screen" , "read" , "nop" et "save". En mode "screen" les données sont affichées à l'écran, le format de ce dernier dépendant du type de données collectées. En mode "save" les données sont écrites dans un fichier pour une utilisation ultérieure. Les deux autres modes sont d'un intérêt moins important pour l'administrateur. Le mode par défaut est le mode "screen".

Les options précédées du signe "-" sont les suivantes :

- help : donne un court listing d'aide
- sleep # : # est le nombre de secondes durant lesquelles le logiciel doit "s'endormir" entre deux prises de mesures consécutives. En mode d'affichage

"save" la valeur par défaut est 10 minutes, autrement il dépend de la vitesse du terminal et varie entre 2 secondes (pour les terminaux dont la vitesse de transmission > 9600 bps) et 8 secondes (pour les terminaux dont la vitesse est de 300 bps et moins).

- file : "file" étant le nom du fichier que le mode d'affichage "save" doit écrire et le mode "replay" lire .

Plusieurs autres options d'importance moindre existent, et leur description ne ferait que compliquer inutilement les fonctionnalités de l'outil.

L'argument "données" permet de sélectionner le domaine ou le groupe de domaines tel qu'ils ont été définis dans les deux tableaux précédents.

Lorsque Monitor fonctionne en mode écran, des commandes permettent de passer d'un domaine ou d'un groupe de domaine à un autre, d'accéder à l'aide ou encore de quitter l'application.

(b) Avantages

La fonction de sauvegarde/récupération permet, en plus de l'affichage des données à l'écran, de les stocker sur disque pour les récupérer ultérieurement. Une option de départ permet de stipuler le temps entre prise de mesures, le temps par défaut étant 10 minutes. Les déclarations de structures et de constantes pour les fichiers de données sont définies dans le fichier RECORD.H . Ce fichier est inclus dans le lot, pour permettre à ceux qui le désirent de rédiger des programmes spécialisés pour lire ces données.

Chaque écran d'affichage de Monitor offre la possibilité d'accéder à une aide à l'écran grâce à la commande "?". Cette aide est toutefois, selon l'avis même de ses concepteurs, "très limitées" mais elle devrait être étendue dans une version suivante.

La variété des données accessibles est plus importante que le logiciel DxTopCPU vu précédemment. S'ajoute en effet aux mesures assurées par ce dernier des grandeurs propres, telles que le nombre d'utilisateurs connectés ou encore des informations sur les processus.

Les fonctionnalités offertes par ce logiciel sont très vastes et permettent de couvrir plusieurs types de besoins bien particuliers. La fonction de sauvegarde répond en partie aux besoins exprimés par les gestionnaires des différents systèmes présents à l'Institut d'Informatique.

L'absence de graphiques permet à cette application de s'exécuter sur des terminaux classiques sans aucun problème.

(c) Inconvénients

La portabilité du logiciel ne semble pas exemplaire, en effet des versions différentes existent pour tourner sur les versions successives du système d'exploitation Ultrix. Il est donc impossible, sans modifier le code de façon importante, d'implanter cet outil sur des machines tournant sous un "autre Unix", comme cela pourrait être le cas à l'Institut.

Le manque de documentation quant à la définition précise des données collectées est un inconvénient grave à notre avis, cette absence peut en effet être à la base d'une mauvaise interprétation des mesures par l'administrateur.

Les nombreuses fonctionnalités offertes par cette application ont un effet pervers immédiat, elles rendent le logiciel compliqué, ce qui veut dire difficile à apprendre, à

comprendre et très peu convivial. Cette caractéristique risque de démotiver l'utilisateur. Une démotivation aurait pour conséquence quasi immédiate de faire chuter la fréquence d'utilisation de l'outil et donc son intérêt.

La complexité réside non seulement dans l'accès à l'application via une ligne de commandes kilométrique mais aussi dans le format du fichier de sauvegarde des données. Cette sauvegarde constitue un argument sans conteste primordial, mais son principe de fonctionnement d'une part et sa complexité d'autre part constitue un inconvénient majeur nous semble-t-il.

Le principe de fonctionnement de cette sauvegarde repose sur une simple déviation de la sortie standard vers un fichier de nom stipulé en argument. Cette déviation ne calcule aucune moyenne et n'est organisée sur aucune synchronisation logique pouvant avoir un intérêt pour l'administrateur du système. Aucune notion temporelle n'est en effet présente, les mesures s'accumulent séquentiellement sans plus, la réutilisabilité n'est donc envisageable que pour une courte période.

Non seulement le principe est à notre avis trop simple, mais la technique de sauvegarde est quant à elle, trop complexe. La sauvegarde se fait sous un format très complexe qui, même s'il est décrit en détails dans les fichiers de documentation, rend très difficile voire impossible la réutilisation des données dans une autre application. Cette réutilisation doit de plus être entièrement conçue par l'utilisateur de manière conforme au format imposé par Monitor. L'utilisation différée, si elle est possible et plus simple, n'en est pas moins soumise à un scénario d'utilisation particulier, passant par des programmes filtres qui permettent de relire les mesures.

L'absence de graphisme, s'il est un avantage dans la question de la portabilité de l'application, n'en reste pas moins un inconvénient majeur en matière de présentation des résultats. Nous avons déjà évoqué pour l'outil précédent l'avantage d'une présentation graphique et ses dangers.

3.2.3. CHOIX D'UNE SOLUTION

Nous avons passé en revue les différentes mesures qui pourraient décrire l'état d'un système informatique. Les outils qui pourraient réaliser leur collecte ont également fait l'objet d'une comparaison. Il reste maintenant à faire un choix quant aux données et aux logiciels qui aideraient le mieux l'administrateur dans sa tâche.

Les avantages et inconvénients des moniteurs matériels et logiciels ont été présentés au point précédent. Nous avons aussi évalué les faiblesses des outils disponibles actuellement au sein de l'Institut. Notre objectif est maintenant de proposer une solution qui pallierait ces inconvénients .

Notre choix va se porter sur le développement d'un moniteur logiciel pour des raisons évidentes. Tout d'abord, ce type d'outil est très facilement modifiable; l'ensemble des mesures collectées peut, par exemple, être étendu à d'autres données dont l'intérêt n'était pas apparu au départ. Ensuite, il est plus adapté pour la prise de mesure du type d'informations qui nous intéressent. Enfin, nous ne possédons pas les connaissances techniques nécessaires à l'implantation d'un outil matériel.

Nous avons décidé de mettre au point un outil réunissant un maximum d'avantages des outils étudiés au point précédent et un minimum d'inconvénients. Il s'agira d'un logiciel de prise de mesure à intervalle prédéfini et de sauvetage de ces mesures sur fichier texte classique. Ainsi l'accent sera mis sur les points suivants :

- Principe de fonctionnement similaire à celui d'un démon système collectant les mesures avec un certain intervalle de temps et sauvegardant les résultats sur disque.
- Simplicité d'emploi dans le but de raccourcir la période d'acquisition et d'augmenter la propension à son utilisation.
- Convivialité maximale
- Portabilité maximale sur tout système Unix ou "personnalisé"
- Documentation suffisante pour permettre une acquisition rapide et une interprétation exacte des mesures.

- Affichage graphique garantissant une analyse correcte des résultats.
- Réutilisabilité maximale et simple des données sauvegardées.
- Variété de mesures répondant aux besoins ressentis sans inonder l'administrateur de grandeurs inutiles ou inopportunes.

Des contacts avec les différents gestionnaires du réseau nous ont permis de cerner les mesures indispensables à une bonne gestion du système, celles-ci ont déjà été présentées. Nous pouvons à présent dresser la liste, par domaines, des données qui seront recueillies par le moniteur, pour autant que leur collecte soit techniquement possible.

Ces mesures sont pour chaque hôte :

Occupation des CPU :

Le pourcentage d'utilisation du processeur dans les trois états suivants : système, utilisateur et endormi .

Le nombre de processus en exécution, en attente et bloqués.

Le nombre d'interruptions, d'appels au système et de changements de contexte du processeur.

La charge moyenne du système (load average) .

Accès au(x) disque(s) :

Pour chaque(s) disque(s) éventuel(s) de l'hôte :

Le nombre d'opérations d'accès au disque par seconde .

Le temps moyen d'accès au disque .

Le nombre de Kbytes transférés par seconde .

Le taux d'occupation du disque .

Utilisation de la mémoire :

Taille de la mémoire utilisée .

Taille de la mémoire libre .

Activité de pagination :

- Le nombre de défauts de page .
- Le nombre de pages en entrée .
- Le nombre de pages en sortie .
- Le nombre de pages libérées .
- Le temps moyen séparant deux défauts de page .
- Le temps moyen séparant deux entrées de page .

Accès au réseau :

- Le nombre de paquets entrant sur le réseau .
- Le nombre d'erreurs lors des entrées sur le réseau .
- Le nombre de paquets entrant dans l'hôte .
- Le nombre d'erreurs lors de ces entrées dans l'hôte .
- Le nombre de collisions détectées par l'hôte .

3.3. DEVELOPPEMENT DE L'OUTIL DE MESURE

3.3.1. PHILOSOPHIE DE L'OUTIL

Comme nous l'avons présenté dans l'analyse théorique, le but du logiciel à développer est de fournir à l'administrateur du réseau, tel qu'il a été présenté dans le chapitre 1, un outil de mesure de l'activité des différents composants du système. L'utilisateur de ce logiciel, ayant une fonction principale de direction et de gestion, doit pouvoir l'utiliser fréquemment, facilement et efficacement à fin d'en tirer un profit évident dans le cadre d'une saine gestion des ressources dont il dispose. Cette particularité de l'utilisateur a des implications importantes sur la conception de l'outil, notamment au niveau des mesures à prendre, de la politique de prise de ces mesures et de l'interface de présentation des résultats, ainsi que des possibilités de réutilisation de ces derniers.

Pour être utilisées de façon rentable par l'administrateur du système, les mesures qui lui sont fournies doivent répondre à plusieurs critères. Ces mesures doivent tout d'abord être correctes en ce sens qu'elles doivent être une image aussi proche que possible de la réalité. Une définition précise de ce que nous entendons par "aussi proche que possible de la réalité" sera donnée dans le cahier des charges de l'outil de mesure. Une qualité indispensable de ces mesures doit être la pertinence. Ce critère permet d'éviter un excès de données peu utiles et de garantir une interprétabilité aisée des résultats conduisant à un diagnostic le plus précis et le plus efficace possible de l'état réel du système.

La politique de prise de mesures doit impérativement respecter les critères définis au paragraphe précédent et principalement la qualité de correction des mesures prises. Comme nous l'avons suggéré au chapitre 2, la philosophie de cet outil sera basée sur la surveillance continue et le stockage permanent des résultats obtenus. Tel le **Holter Monitoring** du cardiologue, ce logiciel sera le témoin permanent, pour l'administrateur, de l'état de santé général du réseau.

Cette constance de la surveillance amène néanmoins à faire plusieurs remarques. Tout d'abord un suivi continu au sens strict du terme n'est pas envisageable dans une configuration telle que nous la connaissons. Elle aurait pour conséquence inéluctable de surcharger inutilement et artificiellement le système et donc de donner des mesures erronées de par son fonctionnement propre. Il sera donc nécessaire de définir un intervalle d'observation qui permette d'une part d'éviter l'écueil de la surcharge artificielle du réseau et d'autre part de garantir une correction et une représentativité des mesures. Une seconde remarque que nous voulons évoquer concerne l'intervalle de temps utilisé lors de la présentation des mesures. Cet intervalle doit d'une part être calqué sur les notions de temps communément admises dans la vie courante (heure, jour, semaine et mois), et d'autre part répondre à un critère d'efficacité et de cohérence de l'interprétation des résultats et donc du diagnostic.

L'interface-utilisateur de l'outil de mesure doit être réduit à la seule présentation des données. L'administrateur du système doit pouvoir considérer le mesureur comme un fantôme lui fournissant des résultats à sa demande, qu'il peut représenter à sa guise et qu'il peut éventuellement réutiliser dans d'autres applications ou dans un rapport quelconque. Cette condition a donc des implications importantes sur la logique de conception du logiciel, d'un côté la partie active de prise de mesure doit être totalement autonome et indépendante et, d'un autre côté, le module de présentation des résultats doit pouvoir être paramétré par l'utilisateur et être d'un usage agréable et convivial. Il est donc nécessaire d'assurer l'interface entre ces deux modules dont tant le principe de fonctionnement que le but sont totalement différents. Pour garantir les qualités que nous venons de décrire, les deux modules composant le système doivent pouvoir s'exécuter sur des machines différentes, chacune appropriée à la finalité du module qu'elle supporte.

L'utilisateur doit pouvoir jouer un rôle essentiel dans le choix tant des données à sélectionner que des modalités de leur présentation. A cet effet, il convient de garantir à l'utilisateur un large choix de possibilité d'affichage des mesures qu'il a sélectionnées (affichage à l'écran, impression sur papier, représentation graphique, présentation en tableau, ...).

L'outil doit permettre une réutilisabilité non seulement des données brutes mais aussi des résultats émanant des différents choix de présentation sélectionnés par l'utilisateur. Cette possibilité sous-entend un stockage ou une impression de ces résultats. Il peut être également envisageable de transférer ces résultats vers une autre application.

3.3.2. CAHIER DES CHARGES

La philosophie du logiciel ayant été abordée au point précédent, nous allons évoquer, dans cette seconde approche, l'aspect réellement technique du produit. Ce cahier des charges sera un relevé des différents critères techniques et pratiques auxquels l'outil devra répondre, sans toutefois correspondre à des spécifications rigoureuses qui seront elles abordées dans la partie consacrée au développement proprement dit du logiciel.

3.3.2.1. MODULE DE PRISE DE MESURES

3.3.2.1.1. Données collectées

Les mesures collectées et décrites ci-dessous le seront pour chaque hôte sur le réseau ayant déjà fait l'objet d'une description dans le chapitre 1.

Les mesures concernant les stations de travail connectées à un même serveur seront regroupées avec les mesures concernant ce dernier.

Les mesures prises seront les suivantes :

1. OCCUPATION DU CPU

- Le pourcentage d'utilisation du processeur dans les trois états suivants : système, utilisateur et endormi.
- Le nombre de processus en exécution, en attente et bloqués.

- Le nombre d'interruptions, d'appels au système et de changements de contexte du processeur (système ou utilisateur).
- La charge moyenne du système (load average) durant la dernière minute, les cinq dernières minutes et les quinze dernières minutes.

2. ACCES AU(X) DISQUE(S)

Ces mesures ne seront bien sûr prises que pour les hôtes possédant un ou plusieurs disques.

Les mesures suivantes seront identiques pour chaque disque:

- Le nombre d'opérations d'accès au disque par seconde.
- Le temps moyen par accès au disque.
- Le nombre de Kbytes transférés par seconde.

En plus de ces mesures, il sera donné le taux d'occupation des partitions suivantes :

/

/usr

/var

/tmp

/users

/users/students

3. UTILISATION DE LA MEMOIRE

- Taille de la mémoire virtuelle utilisée.
- Taille de la mémoire libre.

4. ACTIVITE DE PAGINATION

- Le nombre de défauts de pages.
- Le nombre de pages en entrée.
- Le nombre de pages en sortie.
- Le nombre de pages libérées.
- Le temps moyen séparant deux défauts de page.
- Le temps moyen séparant deux entrées de page.

5. ACCES AU RESEAU

- Nombre de paquets entrant sur le réseau.
- Nombre d'erreurs lors des entrées sur le réseau.
- Nombre de paquets entrant dans l'hôte.
- Nombre d'erreurs lors de ces entrées dans l'hôte.
- Nombre de collisions détectées par l'hôte.

3.3.2.1.2. Stockage des résultats

Les résultats à la sortie du module de prise de mesures devront être sauvegardés dans quatre fichiers. Les deux premiers contiendront les mesures de la journée en cours et de la journée précédente, alors que les deux autres contiendront les mêmes mesures pour le mois courant et le mois précédent. Ces fichiers devront avoir un nom significatif afin de faciliter leur utilisation ultérieure. L'unité temporelle pour les deux fichiers se rapportant aux mesures d'une journée sera l'heure, alors que pour les deux fichiers reprenant les mesures d'un mois, cette unité sera la journée.

3.3.2.1.3. Fréquence des mesures

Afin de garantir une bonne correction des résultats et leur interprétabilité, il devra être procédé à plusieurs prises de mesures par heure. Les différentes prises de mesures durant chaque heure ne seront jamais instantanées de manière à éviter les résultats moyens faussés.

3.3.2.2. MODULE DE PRESENTATION DES RESULTATS

3.3.2.2.1. Sélection des résultats

L'utilisateur devra pouvoir sélectionner en combinaison :

- La période de référence : jour courant, mois courant, jour précédent ou mois précédent.

- Le domaine de mesure désiré : Occupation du CPU, Accès au(x) disque(s), Utilisation de la mémoire, Activité de pagination ou Accès au réseau.

Pour chaque domaine, l'utilisateur doit pouvoir choisir une des mesures telles qu'elles sont définies au point précédent. Le logiciel doit offrir un maximum de combinaisons possibles entre mesures d'un même domaine.

- Le groupe de machines connectées autour d'un même serveur (exemple : electre, ws1, ws2, ws3 et ws4). Pour chaque groupe, l'utilisateur devra choisir un hôte particulier.

3.3.2.2.2. Sélection du type de présentation

Outre la sélection des mesures, l'utilisateur doit pouvoir choisir le type de présentation qui lui semble adéquat. Le choix sera laissé entre une présentation en tableau avec titre et encadrement et une présentation graphique. Il sera proposé à l'utilisateur suffisamment de types de graphiques (au moins la courbe et l'histogramme) pour lui permettre de choisir le plus adapté aux données sélectionnées. Tant pour le tableau que pour les graphiques, la possibilité d'obtenir une moyenne des résultats pour la période sélectionnée devra être offerte, ainsi que celle d'ajouter des titres voir même des commentaires.

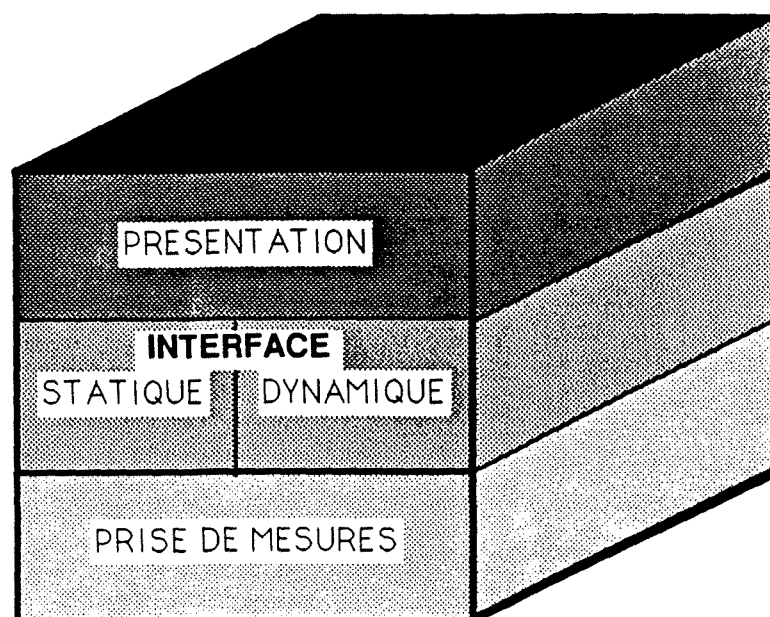
3.3.2.2.3. Réutilisation des résultats

L'utilisateur du logiciel aura la faculté de sauvegarder et d'imprimer aussi bien les tableaux que les graphiques. L'utilisateur doit pouvoir récupérer le résultat dans une application courante d'édition de textes.

3.3.3. DEVELOPPEMENT

Les fonctionnalités à remplir par l'outil de prise de mesures et par l'outil d'affichage ont été clairement précisées dans le point précédent. L'étape de développement de ces deux logiciels peut maintenant être abordée. Notre but, dans ce paragraphe, n'est pas de donner une description précise et rigoureuse de l'implémentation de ces deux programmes, mais plutôt de présenter les décisions importantes qui ont été prises au cours de cette phase. Ces choix feront d'autant plus l'objet de notre attention qu'ils auront un impact sur la qualité de l'information apportée à l'utilisateur.

Le schéma qui suit décrit la première découpe en couche du logiciel. La couche de prise de mesures correspond au module de prise de mesure tel que nous l'avons décrit dans le cahier des charges. La couche interface se subdivise en deux parties, l'interface statique qui est assurée par les fichiers contenant les données à présenter et l'interface dynamique, seule partie que nous n'avons pas développée et qui est un logiciel de transfert de fichiers des machines Unix vers un Macintosh. Le logiciel utilisé est Kermit, répandu et utilisé sur pratiquement toutes les configurations présentes à l'Institut d'Informatique.



3.3.3.1. CHOIX DE DEPART

Nous avons déjà évoqué la possibilité d'implanter le logiciels en deux parties distinctes sur des machines différentes. Cette idée nous donne une liberté de choix tant dans la configuration matérielle que logicielle devant être utilisée pour l'implémentation de chaque module.

3.3.3.1.1. Module de prise de mesures

Le module de prise de mesure doit impérativement résider sur un hôte connecté au réseau, le choix de ce dernier reste encore à déterminer de façon définitive. Nous avons fait un choix arbitraire de départ en décidant d'implémenter le mesureur sur le serveur Electre. Ce choix a été guidé par l'objet initial du mémoire à savoir les nouvelles stations Vax et leur environnement Ultrix.

3.3.3.1.2. Module de présentation

Pour le second module, les critères essentiels qui doivent guider notre choix sont la simplicité, les possibilités graphiques, la convivialité, la disponibilité et la possibilité de réutilisation des résultats.

La simplicité doit garantir que le module de présentation soit maîtrisé moyennant un investissement en temps minimum. Les possibilités graphiques doivent être considérées comme un ensemble de moyens mis à la disposition de l'utilisateur pour représenter un ensemble de valeurs. Cet ensemble de moyens peut aller du simple écran texte à l'outil évolué de construction de graphiques. Nous entendons par convivialité le caractère agréable de l'utilisation de l'outil. Le système supportant le module de présentation est considéré comme disponible si l'administrateur du réseau peut en disposer constamment ou y accéder facilement.

La possibilité de réutilisation invoque automatiquement trois types d'opérations : la sauvegarde, l'impression sur papier de qualité et la récupération par une application courante d'édition de textes. Vu l'utilisation quasi exclusive par les secrétariats, des micro-ordinateurs de type PC et surtout Macintosh pour ce genre d'applications, il nous paraît indispensable d'envisager la réutilisabilité dans ce sens.

Plusieurs solutions globales (matérielles et logicielles) sont envisageables, répondant chacune de manière différente aux critères que l'on s'est fixé.

(a) Serveur Unix et terminal

Cette solution convient d'un point de vue facilité et disponibilité vu d'une part le grand nombre de terminaux accessibles et d'autre part la possibilité de connexion au Pabx à partir de pratiquement tous les locaux. Elle risque cependant de poser problème pour les autres critères. Les possibilités graphiques d'un terminal sont quasi nulles, il est donc impossible de fournir à l'utilisateur un résultat graphique, si ce n'est en utilisant les caractères accessibles pour tenter de le représenter. Un tel procédé manque non seulement de professionnalisme et de convivialité mais il risque également d'induire en erreur l'utilisateur de ces résultats. En effet l'absence d'échelle correcte pourrait être à la base d'une mauvaise interprétation de l'évolution des données. La convivialité d'un tel matériel reste précaire et la réutilisabilité des résultats sélectionnés, tout en restant possible, est loin d'être évidente. En effet, si la sauvegarde et l'impression laser sont accessibles, la possibilité de voir les résultats réinjectés dans une autre application est plus limitée, l'utilisateur se voyant obligé de transférer les résultats, via le réseau, sur un micro-ordinateur à sa disposition.

(b) Serveur Unix et terminal graphique

La seule différence avec la solution précédente est la possibilité de disposer de moyens graphiques plus évolués mais laissant toutefois toute la responsabilité de la création graphique au programmeur. Il existe toutefois un outil intégré d'interprétation des résultats pouvant être utilisé, mais dont le but principal est l'analyse statistique. Il convient également de remarquer que la disponibilité d'un tel matériel est très limitée.

(c) Station de travail

Si la station de travail convient parfaitement pour ce type d'application, ses possibilités graphiques, tout en étant potentiellement appréciables, sont néanmoins limitées par les logiciels disponibles. En effet leur utilisation comme support du module de présentation nous oblige à utiliser *Xwindows* pour développer l'interface graphique. Lors d'un contact informel avec un utilisateur et programmeur averti, il nous est apparu que le caractère très "bas niveau" des routines disponibles dans cet environnement constitue un obstacle à son utilisation. Les délais dont nous disposons ne nous permettent pas de consacrer plusieurs mois à la seule étude de *Xwindows*. Cette contrainte nous oblige à rejeter cette solution ou à utiliser ce matériel de luxe comme simple terminal. Outre cet obstacle, il est à remarquer que la disponibilité des stations de travail est satisfaisante.

(d) Micro-ordinateurs

Si cette solution paraît de prime abord très différente des précédentes et peu en relation avec l'environnement faisant l'objet des mesures, elle répond néanmoins pleinement aux critères retenus. La disponibilité de ce type de matériel, tant pour le Macintosh que pour le PC, est totalement assurée.

La réutilisabilité des résultats obtenus semble idéale pour les deux types de machines, bien que la propension à l'utilisation des Macintosh dans le domaine de l'édition au sein de l'Institut d'Informatique lui donne un certain avantage. Les autres critères vont non seulement dépendre du type de machine mais aussi de l'environnement logiciel choisi pour développer le module de présentation. L'idéal étant donc de d'opter pour un environnement logiciel connu et apprécié répondant aux critères de convivialité, de simplicité et de possibilités graphiques évoqués. Nous avons finalement opté pour le monde Macintosh et le logiciel Excel, combinaison répondant de façon optimale aux besoins de l'administrateur.

3.3.3.2. DEVELOPPEMENT DES MODULES

3.3.3.2.1. Module de prise de mesures

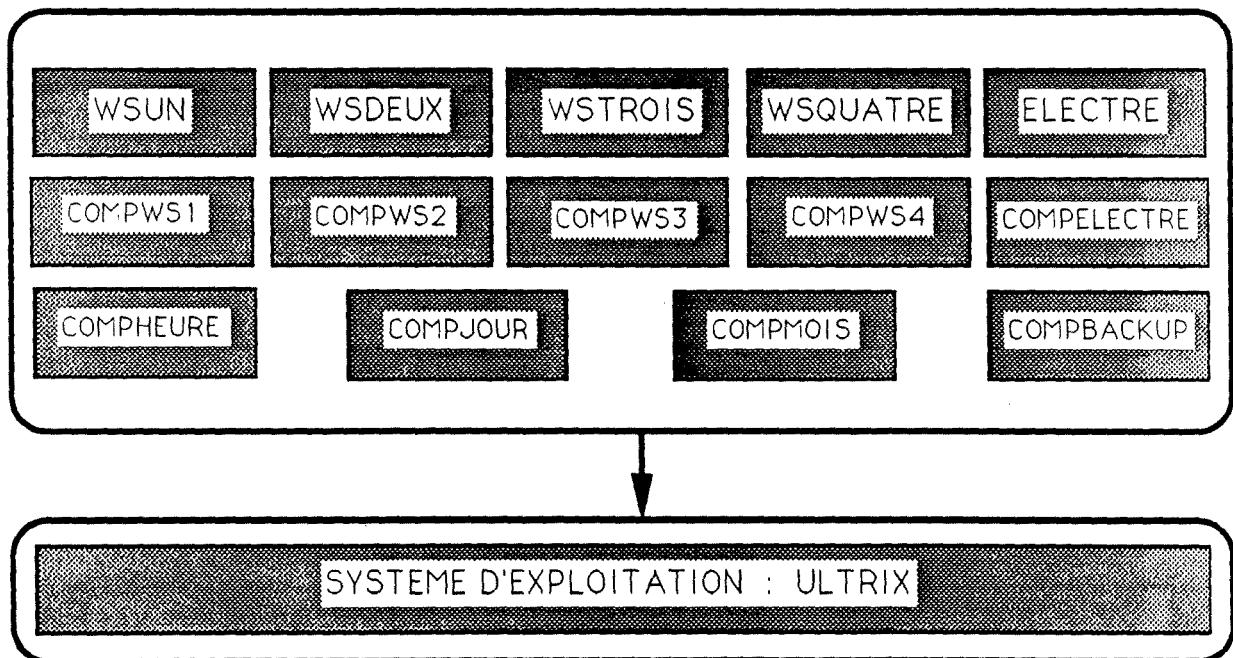
Nous avons suivi pour le développement de ce module une méthodologie inspirée de celle proposée dans le cours de monsieur Van Lamsweerde auquel nous ferons d'ailleurs de nombreuses références. La particularité de ce logiciel nous a néanmoins obligé à adapter considérablement certaines étapes. Nous préciserons tout au long de ce développement les aménagements qui ont été apportés et les particularités qui les justifient.

(a) Conception globale

Nous envisagerons dans cette conception globale l'architecture logique du module de prise de mesure. Les différents modules le composant feront l'objet d'une description dans un second point. Nous aborderons ensuite la justification du choix de cette architecture. Le dernier point traitera des relations logicielles existant entre ces différents composants. Nous terminerons avec les spécifications externes des composants

(1) Architecture

Le schéma qui suit décrit l'architecture logique du module de présentation. Ce dernier se découpe en dix composants, tous regroupés dans une seule couche située juste au dessus du système d'exploitation. Les dix sous-modules utilisent tous des commandes du système Unix, alors qu'ils ne s'utilisent pas entre eux.



(2) Description des modules

module WSUN

Ce module a pour but de collecter et de stocker, pour la station ws1, toutes les informations contenant les mesures définies dans la cahier des charges, en utilisant les outils du système d'exploitation.

module WSDEUX

Ce module a un but identique pour la station WS2.

module WSTROIS

Ce module a un but identique pour la station WS3.

module WSQUATRE

Ce module a un but identique pour la station WS4.

module ELECTRE

Ce module a un but identique pour le serveur Electre, s'ajoutent aux mesures prises sur les stations celles concernant les disques.

module COMPWS1

Ce module filtre les informations récoltées et stockées par le module WSUN, afin d'en extraire les mesures définies dans le cahier des charges et de les stocker sous un format prédéfini.

module COMPWS2

Ce module a le même effet sur les mesures récoltées par le module WSDEUX.

module COMPWS3

Ce module a le même effet sur les mesures récoltées par le module WSTROIS.

module COMPWS4

Ce module a le même effet sur les mesures récoltées par le module WSQUATRE.

module COMPELECTRE

Ce module a le même effet sur les mesures récoltées par le module ELECTRE.

module COMPHEURE

Ce module calcule la moyenne horaire des données récoltées à plusieurs reprises par les modules WS* et ELECTRE et filtrées par les modules COMP*. Ces moyennes sont stockées dans un fichier de format prédéfini et de nom significatif contenant les mesures définies au cahier des charges pour les cinq hôtes et pour chaque heure écoulée de la journée.

module COMPJOUR

Ce module calcule la moyenne journalière des données stockées dans le fichier du jour précédent et la stocke dans le fichier du mois courant.

module COMPMOIS

Ce module stocke les données concernant le mois précédent dans un fichier archive et transfère les valeurs du mois courant dans le fichier du mois précédent.

module COMPBACKUP

Ce module gère le problème des backups du système effectués tous les lundi matins.

(3) Justification de l'architecture

L'architecture logique globale est entièrement basée sur le concept de la synchronisation des différents modules ainsi que sur le principe de l'indépendance de ces mêmes modules. La structure modulaire est une technique qui offre d'une part une factorisation plus grande du travail, et d'autre part une aide pour la maintenance ultérieure du logiciel.

Il nous semble utile de rappeler qu'une structure est modulaire si les composants qui la forment et les relations qui existe entre eux sont tels que les attributs de chaque composant peuvent être défini de manière simple et précise, et si, de plus, chaque composant jouit de trois propriétés :

- capacité maximale à cacher de l'information
- faible degré de couplage
- forte cohésion interne [VAN, 49]

Les attributs des composants sont ses spécifications. Celles-ci ne sont rien d'autre qu'un énoncé de ce que ce module doit faire, son mode de réalisation (algorithme), son procédé de construction c'est-à-dire sa trace (explication détaillée du raisonnement suivi pour construire l'algorithme à partir de ses spécifications), et enfin ses performances mises en exergue par la complexité de son algorithme en temps et en place.

Nous nous attarderons plus spécialement sur les trois propriétés évoquées précédemment et que doit revêtir tout bon module.

Capacité à cacher de l'information

Idéalement chaque module devrait renfermer un maximum de secret sur les décisions de conception qui ont été prises. Ces secrets peuvent concerner le mode de représentation des données (exemple : le format de structure de données intermédiaires), le mode de mémorisation d'une

structure de données ou encore le mode de réalisation d'un traitement (exemple : comment se fait le filtrage des données).

Faible degré de couplage

Le degré de couplage est une mesure de la complexité de l'interface. Cette mesure représente le nombre de connexions statiques de ce module avec d'autres et la complexité de ces connexions. C'est donc une mesure d'interdépendance des modules [VAN,49].

Forte cohésion interne

Le couplage était une propriété inter-module, la cohésion, quant à elle, est une propriété intra-module. C'est une mesure d'interdépendance entre les constituants de ce module. On parlera de cohésion procédurale signifiant que le module réalise un minimum de traitements non liés [VAN,49].

La conséquence évidente du respect de ces propriétés est double, d'une part une réutilisabilité accrue du logiciel et d'autre part une dimension réduite des modules.

Nous allons justifier, pour chaque module de l'architecture, le choix de leur existence propre et vérifier le respect des trois propriétés de base évoquées précédemment.

modules WS* et ELECTRE

Ces modules, dont le rôle est de prendre les mesures et de stocker les premiers résultats bruts dans un fichier, sont quasi identiques mais s'exécutent sur des hôtes différents. Une petite différence existe cependant entre les modules WS* et le module ELECTRE, ce dernier devant prendre des mesures supplémentaires sur les disques dont dispose le serveur contrairement aux stations. L'intérêt d'une telle modularisation est donc double.

Tout d'abord garantir la notion de synchronisation sur laquelle est basée l'architecture du logiciel. Cette synchronisation nécessite un déclenchement simultané de ces cinq modules afin de collecter des mesures plus proches de la réalité, comme nous le montrerons en détail ultérieurement.

Le second intérêt repose sur le besoin de garantir une maintenabilité et une extensibilité aisée du logiciel. Effectivement, le retrait ou l'ajout d'un hôte ne perturbe en rien les modules de prise de mesures sur les autres hôtes. Cette qualité va même plus loin en permettant des prises de mesures sur des hôtes dont les outils ou les commandes système sont différents. Tel est d'ailleurs le risque ou plutôt l'avantage de tirer parti des "améliorations" apportées par les constructeurs à leur version personnalisée du système Unix (SunOS, Ultrix, HP-UV, ...). Nous pouvons prendre en exemple la commande **ETHERFIND** du système d'exploitation des stations de travail Sun qui donne une série de renseignements sur les accès au réseau Ethernet (annexe E).

Les trois qualités que doit comporter un module sont présentes de manière évidente dans ces cinq composants. la capacité de cacher de l'information est grande, la seule chose à savoir c'est qu'ils accèdent au système pour en extraire des données qu'ils stockent dans un fichier de format et de nom connu.

Le faible degré de couplage de l'interface de ces modules est garanti par le fait que la seule connexion statique est le fichier de sortie. La complexité de cette connexion est donc réduite à la connaissance du format constant de ce fichier. La forte cohésion interne repose sur le fait qu'un seul type de traitement est exécuté au sein de ces modules, à savoir l'utilisation de routines systèmes.

module COMPWS* et COMPELECTRE

Ces modules, dont le but est de filtrer les informations réellement demandées à partir des données brutes, sont quasi identiques à la différence près qu'ils ont en interface des collections de données différentes. Le choix d'en faire cinq modules distincts repose non seulement sur cette différence mais aussi sur plusieurs autres facteurs.

Premièrement, la différence est plus marquée pour le module COMPELECTRE, car la collection des données brutes contient plus d'informations, le serveur disposant de deux disques alors que les stations n'en ont pas.

Deuxièmement, cette modularisation accroît considérablement la maintenabilité et l'extensibilité du logiciel dans le même sens que celui évoqué pour les cinq premiers modules. Ce choix permet en effet de modifier le format ou le contenu des données brutes pour un hôte, sans pour autant revoir le module se rapportant aux autres hôtes. Ce serait par exemple le cas si on décidait d'ajouter une unité de disque à une des stations. Cette optique admet également une modification de la configuration, sans occasionner de bouleversement du logiciel. On peut prendre comme exemple le cas de l'intégration d'un ou plusieurs nouveaux hôtes dans le réseau.

Troisièmement, cette découpe permet d'assurer la continuité de fonctionnement du logiciel en parallèle avec sa maintenance et son extension. L'adaptation de la configuration, dans un des sens évoqué au point précédent, ne met nullement en péril le fonctionnement des autres modules et garantit par là même la disponibilité des mesures pour l'administrateur du réseau.

Enfin cette décomposition du logiciel autorise une portabilité qui est accrue par la diversification autorisée de ses composants de base et donc des formats de données brutes, ainsi que par la facilité d'adaptation des algorithmes. Cette portabilité est une qualité importante pour ce type d'outil et

particulièrement dans un environnement basé sur un réseau reliant des machines hétérogènes mais fonctionnant sous des systèmes d'exploitation semblables sinon tout au moins très proches.

Le respect des propriétés garantissant une bonne structuration semble évident. La capacité de cacher de l'information est grande puisque toute la liberté est laissée au concepteur pour réaliser son filtrage.

Le degré de couplage reste faible vu que la seule connexion statique présente dans l'interface de ces modules est la collection de données brutes en amont et les données filtrées en aval. La complexité de ces connexions est raisonnable vu le format standard et constant des deux ensembles de données.

Enfin la cohésion interne est très forte car ces modules réalisent un seul type de traitement.

module COMPHEURE

Ce module, dont le but est de calculer une moyenne sur les valeurs filtrées de plusieurs mesures, se doit de par ce rôle de constituer un module à lui seul. Sa modularisation indépendante, par rapport aux modules vu précédemment, repose sur deux caractéristiques de base de l'architecture.

Tout d'abord l'idée de synchronisation nous oblige à prévoir des séries d'exécutions simultanées des modules WS* - COMPWS*, le nombre de répétitions de ces séries restant à déterminer. Cette suite doit être suivie du déclenchement horaire du module COMPHEURE. Cette contrainte temporelle de déclenchement nous impose une conception totalement indépendante du module de calcul horaire.

Cette conséquence est renforcée par une seconde caractéristique, la nécessité de garantir une maintenabilité et

une adaptabilité aisée du logiciel. Il est en effet très facile, dans une architecture telle que nous l'avons proposée, de modifier le nombre de séries d'exécution des modules WS* et COMPWS*, le nombre de prises de mesures et de filtrages primaires n'ayant qu'un effet réduit et facilement appréhendable sur le module COMPHEURE.

Les qualités définies au début de ce point sont présentes de manière évidente dans ce module.

Il est clair que sa faculté de cacher de l'information réside dans les mêmes principes que ceux évoqués pour les modules précédents.

Le degré de couplage est déterminé par l'interface constante de ce module avec les cinq modules en amont et par l'interface avec le module COMPJOUR en aval, cette dernière étant constituée par une seule collection de données de format constant.

Une caractéristique importante est à relever concernant ce module, nous pouvons en effet déjà le considérer comme un module terminal en ce sens que le résultat de sa réalisation est déjà un résultat final du logiciel tout en étant une donnée pour un autre module.

Sa forte cohésion interne repose sur la même propriété que celle évoquée pour les modules précédents, à savoir la réalisation d'un seul type de traitement.

module COMPJOUR

Le but de ce module est similaire au précédent mais dans un contexte temporel différent et avec un moment et une fréquence de déclenchement différents. Les remarques soulevées pour le module COMPHEURE peuvent donc s'appliquer au module COMPJOUR.

Le respect des qualités évoquées tout au long de ce point est garanti selon les mêmes principes que ceux mis en exergue pour le module COMPHEURE.

module COMPMOIS

Un raisonnement identique peut être tenu également pour ce module qui diffère des deux précédents par son contexte temporel de réalisation.

module COMPBACKUP

Ce module a un but totalement extérieur à l'ensemble de synchronisation des modules déjà vus. Il répond à un problème auxiliaire, celui des **backups** du premier jour ouvrable de chaque semaine. La nécessité d'en faire un module indépendant tombe sous le sens pour plusieurs raisons.

Premièrement sa finalité est totalement différente des autres modules, il n'entre pas dans la chaîne de mesure telle que nous l'avons présentée. Le choix d'en faire un module indépendant garantit donc sa forte cohésion interne et son faible degré de couplage avec les autres modules. Aucune interface n'existe en amont et seul le résultat des mesures journalières constitue son interface en aval.

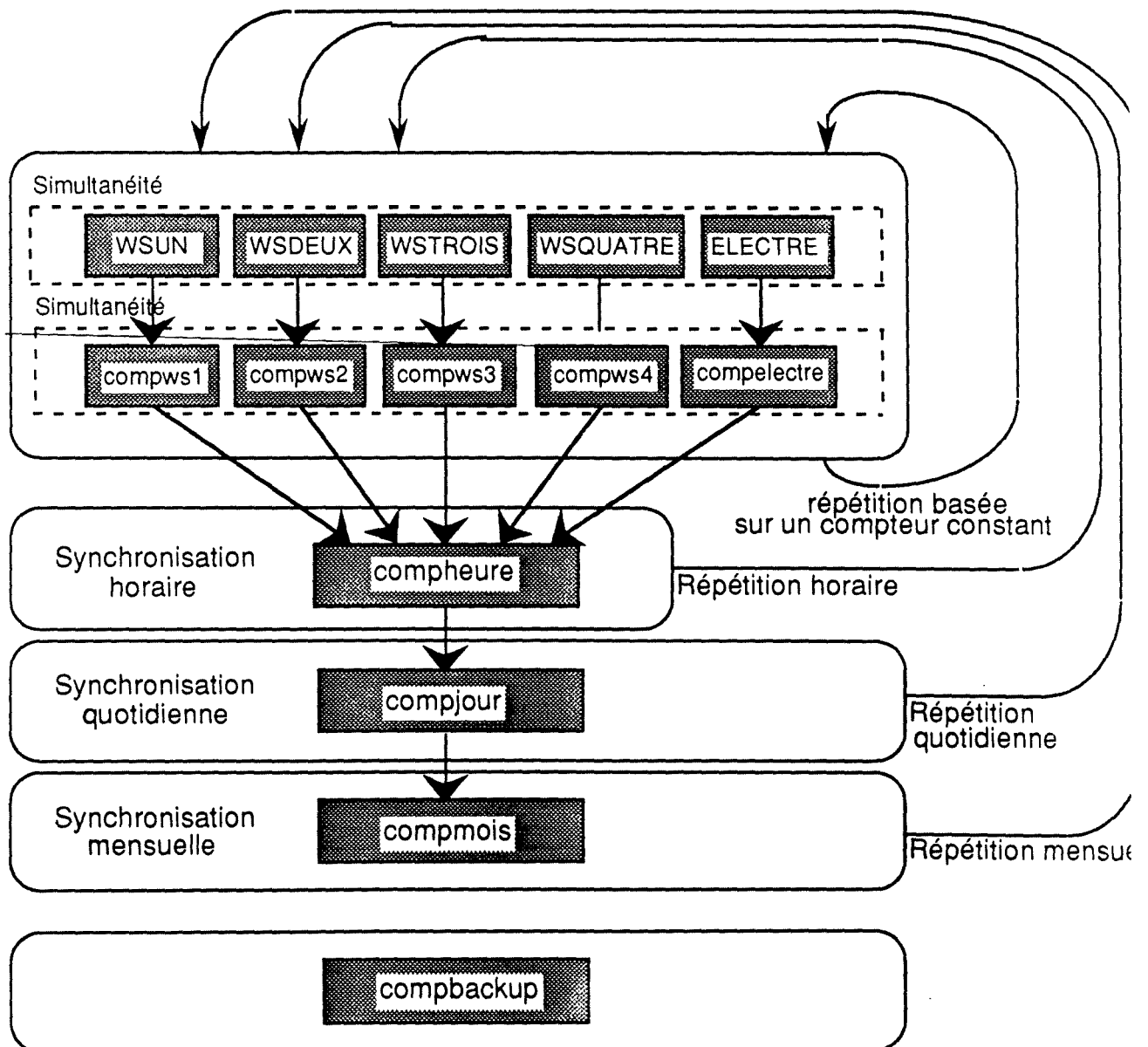
Deuxièmement, il a un caractère "asynchrone" dans l'ensemble. Son rôle est de briser la suite logique afin de garantir la cohésion et la cohérence des résultats.

Enfin, il permet de garantir la troisième qualité, la capacité à cacher de l'information, en masquant complètement, au concepteur des autres modules, le problème de la gestion des interruptions pendant le backup.

Coordinateur

Après avoir justifié la décomposition module par module, il nous semble nécessaire de positionner clairement ce contexte de synchronisation temporelle sur lequel repose toute

l'architecture. Cette synchronisation joue en fait le rôle du coordinateur tel qu'il apparaît dans le cours de méthodologie de développement de logiciel. Ce coordinateur n'apparaît pas encore clairement au niveau de la conception logique globale, mais prendra une forme plus précise lors de la conception physique du logiciel. Toutefois le schéma qui suit permet de visualiser cette logique de coordination et de comprendre encore mieux les choix faits d'emblée.



(4) Relation logicielle

Relation UTILISE

La relation utilise peut être définie de la manière suivante :

A utilise B si et seulement si la correction de A dépend de la disponibilité d'une version correcte de B.

Cette relation existe entre tous les modules décrits au point précédent et le système d'exploitation. Chaque module développé utilise effectivement les primitives du système d'exploitation.

Aucune autre relation n'existe entre les modules définis dans l'architecture.

(5) Spécifications externes

Spécifier correspond à définir de façon complète, précise et rigoureuse le problème; c'est-à-dire à exprimer ce que le module doit réaliser. La nécessité de spécifier se base sur le principe qui veut qu'un module fasse exactement ce que l'on veut, si l'on sait soi-même exprimer clairement ce qu'il doit faire [VAN,49].

Les spécifications rigoureuses sont surtout nécessaires pour les gros logiciels, mais sont toujours considérées comme un avantage important pour les logiciels de petite taille. Cette nécessité de spécifier repose sur plusieurs facteurs.

L'utilisateur est presque toujours une personne différente du concepteur et ce, même pour les petits logiciels.

Le responsable de la maintenance, s'il est souvent le réalisateur pour les petits logiciels, est quasi toujours une personne différente pour les plus grands.

Les coûts de modification, s'ils restent raisonnables pour les logiciels de petite taille, sont souvent trop élevés pour les grosses applications.

Toutes ces caractéristiques nous imposent donc de bien spécifier notre outil de mesure qui, tout en étant un logiciel

modeste, doit néanmoins pouvoir être adaptable et maintenable par d'autres personnes que nous-même.

Nous allons, dans ce point, spécifier chaque module de l'architecture selon la technique de spécification par assertion [VAN,49].

module WSUN

arguments : néant
préconditions : néant
résultat : le fichier WSUN.DAT
postconditions : les résultats des commandes Unix "vmstat 1 10", "vmstat -t", "netstat -i" et "w" exécutées sur la station WS1 sont concaténés dans le fichier WSUN.DAT

module WSDEUX

arguments : néant
préconditions : néant
résultat : le fichier WSDEUX.DAT
postconditions : les résultats des commandes Unix "vmstat 1 10", "vmstat -t", "netstat -i" et "w" exécutées sur la station WS2 sont concaténés dans le fichier WSDEUX.DAT

module WSTROIS

arguments : néant
préconditions : néant
résultat : le fichier WSTROIS.DAT
postconditions : les résultats des commandes Unix "vmstat 1 10", "vmstat -t", "netstat -i" et "w" exécutées sur la station WS3 sont concaténés dans le fichier WSTROIS.DAT

module WSQUATRE

arguments : néant
préconditions : néant
résultat : le fichier WSQUATRE.DAT
postconditions : les résultats des commandes Unix "vmstat 1 10", "vmstat -t", "netstat -i" et "w" exécutées sur la station WS4 sont concaténés dans le fichier WSQUATRE.DAT

module ELECTRE

arguments : néant
préconditions : néant
résultat : le fichier ELECTRE.DAT
postconditions : les résultats des commandes Unix "vmstat 1 10", "vmstat -t", "iostat 1 10", "df", "netstat -i" et "w" exécutées sur le serveur ELECTRE sont concaténés dans le fichier ELECTRE.DAT

module COMPWS1

arguments : fichier WSUN.DAT
préconditions : = postconditions de WSUN
résultats : fichier COMPWS1.DAT = { n groupes |
groupe = { (r, b, w, us, sy, id, int, app, cs,
L1, L5, L15, avm, free, re, pi, po, fr, rem,
pim, inp, iner, out, outer, coll) $\in R^+$ et $n \in N^+$ } (annexe J)

postconditions :
- définitions des composants du groupe :
r = le nombre moyen de processus exécutés, cette mesure étant obtenue en calculant la moyenne de 10 prises de mesures (une chaque seconde) consécutives

b = le nombre moyen de processus bloqués, cette mesure étant obtenue en calculant la moyenne de 10 prises de mesures (une chaque seconde) consécutives

w = le nombre moyen de processus en attente, cette mesure étant obtenue en calculant la moyenne de 10 prises de mesures (une chaque seconde) consécutives

us = le pourcentage moyen d'utilisation du processeur en mode utilisateur, cette mesure étant obtenue en calculant la moyenne de 10 prises de mesures (une chaque seconde) consécutives

sy = le pourcentage moyen d'utilisation du processeur en mode système, cette mesure étant obtenue en calculant la moyenne de 10 prises de mesures (une chaque seconde) consécutives

id = le pourcentage moyen d'utilisation du processeur en mode endormi, cette mesure étant obtenue en calculant la moyenne de 10 prises de mesures (une chaque seconde) consécutives

int = le nombre total d'interruptions intervenues depuis la dernière activation de WSUN

app = le nombre total d'appels système intervenus depuis la dernière activation de WSUN

cs = le nombre total de changements de contexte intervenus depuis la dernière activation de WSUN

avm = la taille moyenne de la mémoire virtuelle, cette mesure étant obtenue en calculant la moyenne de 10 prises de mesures (une chaque seconde) consécutives

free = la taille moyenne de la mémoire libre, cette mesure étant obtenue en calculant la moyenne de 10 prises de mesures (une chaque seconde) consécutives

re = le nombre total de défauts de page survenus depuis la dernière activation de WSUN

pi = le nombre total de pages paginées en entrée depuis la dernière activation de WSUN

po = le nombre total de pages paginées en sortie depuis la dernière activation de WSUN

fr = le nombre total de pages libérées par le démon gérant l'horloge du système depuis la dernière activation de WSUN

rem = temps moyen séparant deux défauts de page au moment de la prise de mesure

pim = temps moyen séparant deux pages en entrée au moment de la prise de mesure

inp = nombre total de paquets entrés dans la station WS1 depuis la dernière activation de WSUN

iner = nombre total d'erreurs détectées sur les paquets en entrée de la station WS1 depuis la dernière activation de WSUN

out = nombre total de paquets sortis de la station WS1 depuis la dernière activation de WSUN

outer = nombre total d'erreurs détectées sur les paquets en sortie de WS1 depuis la dernière activation de WSUN

coll = nombre total de collisions détectées par la station WS1 depuis la dernière activation de WSUN

- format du fichier COMPWS1.DAT:

CPU	R	B	W	US	SY	ID	INT	APP	CS	L1	L5	L15
Disque	AVM FREE											
Mémoire	RE	PI	PO	FR	REM	PIM						
Réseau	INP	INER	OUT	OUTER	COLL							
Groupe 1												
▪												
▪												
▪												
Groupe 2												
▪												
▪												
▪												
Groupe n												

module COMPWS2

arguments : fichier WSDEUX.DAT
préconditions : = postconditions de WSDEUX
résultats : fichier COMPWS2.DAT = { n groupes |
groupe = { (r, b, w, us, sy, id, int, app, cs,
L1, L5, L15, avm, free, re, pi, po, fr, rem,
pim, inp, iner, out, outer, coll) $\in \mathbb{R}^+$ et $n \in \mathbb{N}^+$ } (annexe J)
postconditions : - définitions des composants du groupe :
idem mais mesures prises sur la station
ws2
- format du fichier COMPWS2.DAT :
identique à celui de COMPWS1.DAT

module COMPWS3

arguments : fichier WSTROIS.DAT
préconditions : = postconditions de WSTROIS
résultats : fichier WSTROIS.DAT = { n groupes |
groupe = { (r, b, w, us, sy, id, int, app, cs,
L1, L5, L15, avm, free, re, pi, po, fr, rem,
pim, inp, iner, out, outer, coll) $\in \mathbb{R}^+$ et $n \in \mathbb{N}^+$ } (annexe J)
postconditions : - définitions des composants du groupe :
idem mais mesures prises sur la station
ws3
- format du fichier COMPWS3.DAT :
identique à celui de COMPWS1.DAT

module COMPWS4

arguments : fichier WSQUATRE.DAT
préconditions : = postconditions de WSQUATRE
résultats : fichier WSQUATRE.DAT = { n groupes |
groupe = { (r, b, w, us, sy, id, int, app, cs,
L1, L5, L15, avm, free, re, pi, po, fr, rem,
pim, inp, iner, out, outer, coll) $\in \mathbb{R}^+$ et $n \in \mathbb{N}^+$ } (annexe J)
postconditions : - définitions des composants du groupe :
idem mais mesures prises sur la station
ws4
- format du fichier COMPWS4.DAT :
identique à celui de COMPWS1.DAT

module COMPELECTRE

arguments : fichier ELECTRE.DAT
préconditions : = postconditions de ELECTRE
résultats : fichier ELECTRE.DAT = { n groupes |
groupe = { (r, b, w, us, sy, id, int, app, cs,
L1, L5, L15, ops1, bps1, tps1, ops2, bps2,
tps2, base, usr, var, tmp, users, students,
avm, free, re, pi, po, fr, rem, pim, inp, iner,
out, outer, coll) $\in \mathbb{R}^+$ et $n \in \mathbb{N}^+$ }
postconditions :
- définitions des composants du groupe :
définition identique pour les composants déjà présents
dans COMPWS1, pour les autres :
ops1 = le nombre moyen de recherches par seconde sur
le disque 1 du serveur ELECTRE, cette donnée étant
obtenue en calculant la moyenne de 10 prises de
mesures (une chaque seconde) consécutives

bps1 = le nombre moyen de Kbytes transferés par seconde sur le disque 1 du serveur ELECTRE, cette donnée étant obtenue en calculant la moyenne de 10 prises de mesures (une chaque seconde) consécutives

tps1 = le nombre moyen de transfert exécutés par seconde sur le disque 1 du serveur ELECTRE, cette donnée étant obtenue en calculant la moyenne de 10 prises de mesures (une chaque seconde) consécutives

ops2 = le nombre moyen de recherche par seconde sur le disque 2 du serveur ELECTRE, cette donnée étant obtenue en calculant la moyenne de 10 prises de mesures (une chaque seconde) consécutives

bps2 = le nombre moyen de Kbytes transferés par seconde sur le disque 2 du serveur ELECTRE, cette donnée étant obtenue en calculant la moyenne de 10 prises de mesures (une chaque seconde) consécutives

tps2 = le nombre moyen de transfert exécutés par seconde sur le disque 2 du serveur ELECTRE, cette donnée étant obtenue en calculant la moyenne de 10 prises de mesures (une chaque seconde) consécutives

base, usr, var, tmp, users, students = le nombre de des partitions correspondantes

- format du fichier COMPELECTRE.DAT :

identique à celui de COMPWS1.DAT mais une ligne comprenant les composants supplémentaires est insérée entre la première et la deuxième ligne

module COMPHEURE

arguments :	fichiers COMP*.DAT, PERFORMANCE.JC
préconditions :	= les postconditions des modules COMP*, de plus le fichier PERFORMANCE.JC contient les données concernant la journée en cours sous le format défini
résultats :	le fichier PERFORMANCE.JC

postconditions : le fichier PERFORMANCE.JC contient les données concernant la journée (les mesures de la dernière heure y figurant) et respecte le format défini ci-après

		ELECTRE	WS1	WS2	WS3	WS4
L1	CPU ■ ■ ■	R ---- L15				
L24	CPU	R ---- L15				
L1	Disque ■ ■ ■	Ops1 ---- Stud				
L24	Disque	Ops1 ---- Stud				
L1	Mémoire ■ ■ ■	Avm Free				
L24	Mémoire	Avm Free				
L1	Pagination ■ ■ ■	Re ---- Pim				
L24	Pagination	Re ---- Pim				
L1	Reseau ■ ■ ■	Inp ---- Col				
L24	Reseau	Inp ---- Col				

remarque : le fichier PERFORMANCE.MC a un format identique mais comporte 31 lignes par section

module COMPJOUR

arguments : les fichiers PERFORMANCE.JC et PERFORMANCE.MC

préconditions : les fichiers PERFORMANCE.JC et PERFORMANCE.MC contiennent respectivement les données de la journée et du mois courant et respectent le format défini

résultats : les fichiers PERFORMANCE.JP et PERFORMANCE.MC

postconditions : le fichier PERFORMANCE.JC est vide
le fichier PERFORMANCE.JP a le même contenu que PERFORMANCE.JC avant l'exécution
le fichier PERFORMANCE.MC contient les données concernant le mois en cours (les mesures de la dernière journée y figurant) et il respecte le format défini

module COMPMOIS

arguments : les fichiers PERFORMANCE.MC et ARCHIVE

préconditions : le fichier PERFORMANCE.MC contient les données concernant le mois en cours et respecte le format défini
le fichier ARCHIVE contient la concaténation des fichiers PERFORMANCE.MC de chaque fin de mois

résultats : les fichiers PERFORMANCE.MC , PERFORMANCE.MP et ARCHIVE

postconditions : le fichier PERFORMANCE.MC est vide
le fichier PERFORMANCE.MP a le même contenu que PERFORMANCE.MC avant l'exécution

le fichier ARCHIVE contient la concaténation des fichiers PERFORMANCE.MC de chaque fin de mois (les mesures du dernier mois y figurant)

module COMBACKUP

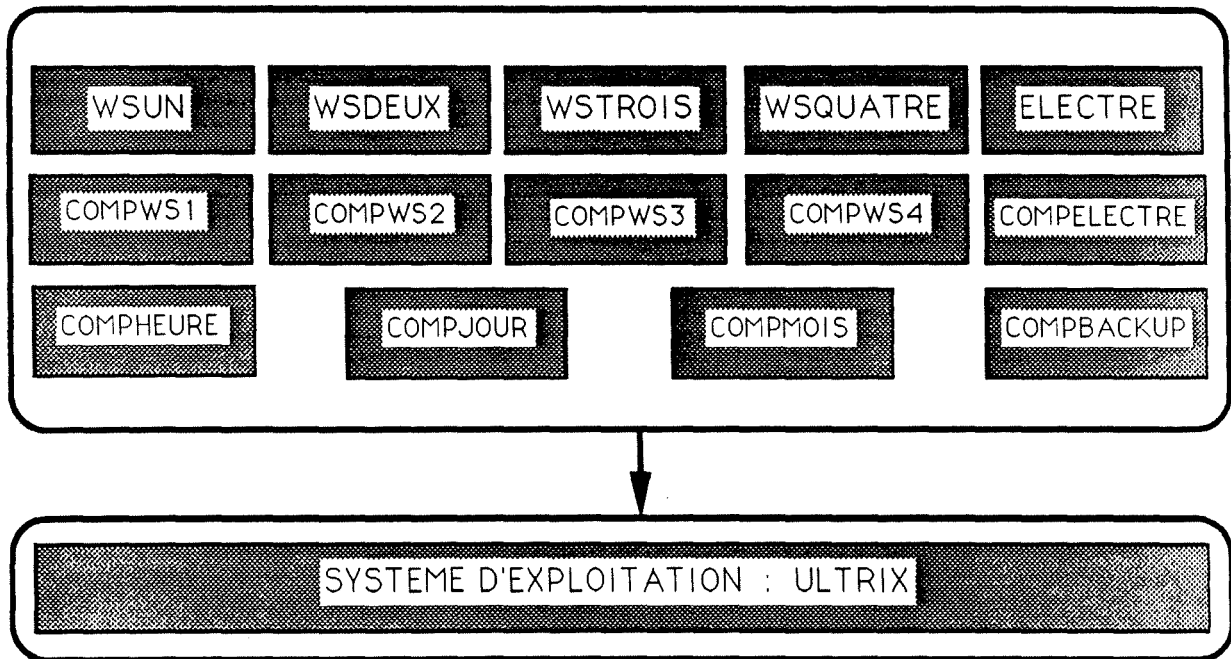
arguments :	le fichier PERFORMANCE.JC
préconditions :	le fichier PERFORMANCE.JC contient les données concernant la journée en cours sous le format défini
résultats :	le fichier PERFORMANCE.JC
postconditions :	le fichier PERFORMANCE.JC contient les données concernant la journée en cours, les mesures concernant les quatre dernières heures étant à zéro

(b) Conception détaillée

Après avoir conçu l'architecture logique globale du logiciel, vient l'étape de la conception physique détaillée de chaque module. Il faut passer des modules logiques aux modules physiques et des relations logiques aux relations physiques. Nous aborderons plusieurs sous-étapes au sein de cette seconde partie du développement, toutefois la méthodologie enseignée a dû être adaptée au cas particulier de ce logiciel. Les spécifications internes ont été enlevées vu la suffisance des spécifications externes, celles-ci ayant été énoncées de manière précise. De même l'algorithme abstrait n'a guère de sens dans ce contexte vu l'utilisation du seul outil de filtrage *awk* du système Unix. Nous nous bornerons à évoquer brièvement l'architecture physique choisie et à mettre au point un plan de tests. Aucun tests *White Box* n'a été prévu puisqu'ils ne pouvaient se baser sur aucun algorithme structuré au sens communément admis. Pour terminer, nous examinerons l'implémentation de chaque module en détail ainsi que la conduite des test individuels.

(1) Architecture physique

Le schéma de l'architecture physique qui suit est en tout point identique à celui de l'architecture logique.



Le choix d'une architecture physique repose sur trois critères essentiels. La traçabilité est sans doute le principal et exprime la qualité de contact entre architecture logique et architecture physique. Etant donné tout changement dans l'architecture logique, on peut aisément retrouver sa place dans l'architecture physique. Les performances représentent le deuxième critère de choix, et bien sûr les contraintes de l'environnement technique tant matériel que logiciel constituent le troisième critère. Paradoxalement les performances de ce logiciel ne représentent pas du tout nos préoccupations principales, en effet, cet outil est un mécanisme synchronisé ne nécessitant aucune interaction. Ses performances au sens communément admis ne nous intéressent donc pas. Il s'agit en fait d'un logiciel totalement exécuté par l'activation synchronisée de petits processus de type "démon".

Nous avons donc basé notre choix sur le souci de garantir une traçabilité maximale tout en restant fidèle aux contraintes imposées par l'environnement.

(2) Plan de tests

S'il est incontestable que l'établissement de plans de test rigoureux est primordial pour un développement efficace d'un logiciel, notre outil de mesure est tellement particulier que seuls des test sur les spécifications peuvent être préparés.

Nous nous bornerons donc à prévoir, pour chaque module, une exécution dans les conditions prévues par les préconditions et vérifier qu'il respecte bien les postconditions au terme de sa réalisation. Certains modules pouvant acquérir des arguments ayant un statut variable, nous envisagerons dans leur cas des test sur les différents cas.

Il reste à noter que certaines préconditions peuvent ne pas être remplies à cause d'un dysfonctionnement du système d'exploitation. Certains cas courant de ce type ont été prévu, tel est le cas du backup, mais d'autres ne l'ont pas été pour des raisons évidentes d'impossibilité de gestion.

module WSUN

Arguments :	Néant
Résultats :	fichier WSUN.DAT
Conditions :	C1 : Les commandes sont bien exécutées et donnent les résultats escomptés. C2 : Aucune commande n'a fonctionné.
Effets :	E1 : Le fichier WSUN.DAT a le format prévu aux préconditions. E2 : Le fichier WSUN.DAT est vide.

Table de décision

	classe normale	classe anormale
C1	X	-
C2	-	X
E1	X	-
E2	-	X
Représentants	Essai avec commandes qui fonctionnent	Essai avec commande qui ne fonctionnent pas (ex:station inaccessible)

modules WSDEUX,WSTROIS,WSQUATRE,ELECTRE

Les différents points repris pour le module WSUN sont entièrement applicables aux modules WSDEUX, WSTROIS, WSQUATRE et ELECTRE. Seuls les noms des fichiers en résultats sont respectivement WSUN.DAT, WSDEUX.DAT, WSTROIS.DAT, WSQUATRE.DAT et ELECTRE.DAT.

module COMPWS1

Arguments : fichier WSUN.DAT
Résultats : fichier COMPWS1.DAT
Conditions : C1 : Le fichier WSUN.DAT a le format prévu aux préconditions.
C2 : Le fichier WSUN.DAT est vide.
C3 : Le fichier WSUN.DAT a un autre format.
Effets : E1 : Le fichier COMPWS1.DAT a le format prévu aux postconditions et contient les valeurs correctes extraites du fichier WSUN.DAT.
E2 : Le fichier COMPWS1.DAT a le format prévu aux postconditions et

contient des valeurs toutes nulles dans le dernier groupe.

E3 : Le fichier COMPWS1.DAT a le format prévu aux postconditions mais contient des valeurs incohérentes (valeurs numériques non correspondantes ou valeurs non numériques).

Table de décision

	classe normale	classe anormale
C1	X	- -
C2	-	X -
C3	-	- X
E1	X	- -
E2	-	X -
E3	-	- X
Représentants	Le fichier WSUN.DAT a bien le format prévu	1. Le fichier WSUN.DAT est vide 2. Le fichier WSUN.DAT contient des valeurs ne répondant pas au format

modules COMPWS2, COMPWS3, COMPWS4 et COMPELECTRE

Tout comme pour les module de prise de mesures de base, on peut appliquer aux modules de filtrage COMPWS2, COMPWS3, COMPWS4 et COMPELECTRE les mêmes points que ceux développés pour le module COMPWS1 à l'exception des noms des fichiers repris en argument et en résultat qui sont adaptés logiquement.

module COMPHEURE

Arguments : Fichiers COMPWS1.DAT
COMPWS2.DAT
COMPWS3.DAT
COMPWS4.DAT
COMPELECTRE

Résultats : fichier PERFORMANCE.JC

Conditions : C1 : tous les fichiers COMP*.DAT ont le format prévu aux préconditions et contiennent des valeurs ≥ 0 .

C2 : Un ou plusieurs des fichiers COMP*.DAT a (ont) un nombre de groupes $< n$, avec n étant le nombre de prise de mesures effectuées par heure.

Effets : E1 : Une ligne vide est enlevée du fichier PERFORMANCE.JC et une ligne des valeurs moyennes par mesure des n prises de mesures effectuées est ajoutée au fichier en respectant le format défini aux postconditions.

E2 : Les valeurs moyennes sont calculées sur $p < n$ prises de mesures et le fichier PERFORMANCE.JC est modifié conformément aux postconditions.

Table de décision

	classe normale	classe anormale
C1	X	-
C2	-	X
E1	X	-
E2	-	X
Représentants	Tous les fichiers comp*.dat ont le format prévu et n'ont pas de groupes dont toutes valeurs = 0	Un ou plusieurs fichiers comp*.dat ont un ou plusieurs groupes dont toutes valeurs = 0

module COMPJOUR

Arguments : Fichier PERFORMANCE.JC

Résultats : Fichiers PERFORMANCE.JC
PERFORMANCE.JP
PERFORMANCE.MC

Conditions : C1 : Le fichier PERFORMANCE.JC est de format normal répondant aux préconditions.

Effets : E1 : Le fichier PERFORMANCE.JC est réinitialisé.
E2 : Le fichier PERFORMANCE.JP est renouvelé.
E3 : Le fichier PERFORMANCE.MC est mis à jour.

Table de décision

	classe normale
C1	X
E1	X
E2	X
E3	X
Représentant	Exécution simple du module

module COMPMOIS

Arguments : Fichier PERFORMANCE.MC

Résultats : Fichiers PERFORMANCE.MC
PERFORMANCE.MP

Conditions : C1 : Le fichier PERFORMANCE.MC est complet.

Effets : E1 : Le fichier PERFORMANCE.MC est réinitialisé.
E2 : Le fichier PERFORMANCE.MP est renouvelé.

Table de décision

	classe normale
C1	X
E1	X
E2	X
Représentant	Exécution simple du module

module COMBACKUP

Arguments : Néant
Résultats : Fichier PERFORMANCE.MC
Conditions : Néant
Effets : E1 : Le fichier PERFORMANCE.JC est mis à jour par l'ajout de quatre lignes de valeurs toutes = 0.

Table de décision

	classe normale
E1	X
Représentant	Exécution simple du module

(3) Implémentation des modules

Nous allons aborder module par module la technique d'implémentation utilisée, le terme technique répondant plus justement à la réalité que celui d'algorithme. Pour chaque module nous développerons, sur base du code repris à l'annexe F, les mécanismes utilisés pour garantir les postconditions imposées.

module WSUN

Le rôle de ce module a déjà été explicitement défini dans un paragraphe précédent, rappelons qu'il doit exécuter plusieurs commandes Unix sur la station WS1, les résultats de ces commandes étant déviées de la sortie standard vers un fichier nommé WSUN.DAT. Le texte du code est disponible à l'annexe G.

Trois points essentiels dans le développement de ce module sont à soulever.

Premièrement, il nous fallait forcer l'exécution, sur la station WS1, des commandes contenues dans le programme shell, alors que le programme en lui-même est exécuté par un processus sur le serveur ELECTRE. Ce problème est à ce niveau facilement résolu grâce au **remote-shell** (commande *rsh [hôte] [commande à exécuter]*). Ce mécanisme permet de créer un **processus shell** sur l'hôte spécifié avec le même nom d'utilisateur et donc sans devoir entrer une nouvelle fois dans le système. Cette technique n'est possible que si on veut accéder à un hôte travaillant dans le même **système de fichier**. Il reste donc à forcer une exécution séquentielle des commandes *vmstat 1 10*, *vmstat -t*, *netstat -i*, *w* sur la station WSUN grâce à un remote-shell en déviant les résultats vers le fichier WSUN.DAT.

Deuxièmement, la constance de l'ordre des commandes est nécessaire pour répondre aux postconditions du module, mais l'ordre en lui-même a un intérêt particulier. L'ordre défini dans les postconditions n'est pas le fruit du hasard, il repose sur la contrainte du format permanent du fichier contenant les résultats. Lorsque l'on sait que l'on utilise l'outil de filtrage *awk* qui travaille en ligne, il est nécessaire que le numéro des lignes contenant les données qui nous intéressent reste constant. Cette nécessité nous oblige donc à placer la commande *W* en fin de module car son résultat est variable en nombre de lignes contrairement aux autres commandes.

Enfin, pour respecter les postconditions, il faut que le fichier soit créé à chaque mesure et donc que l'ancien contenu

soit effacé. La première commande effacera donc l'ancien contenu du fichier en utilisant une déviation de sortie de type "écrasante" (>), alors que les déviations inhérentes aux commandes suivantes seront de type "ajout" (>>).

Les commandes successives sont donc stockées dans un **programme shell**, qui n'est rien d'autre qu'un fichier texte rendu exécutable au moyen de la commande *chmod u+x wsun* déjà évoquée au chapitre 2 consacré aux problèmes de sécurité.

module WSDEUX, WSTROIS, WSQUATRE

La technique évoquée pour le module WSUN est totalement identique pour les modules générant les mesures sur les trois autres stations Vax. Seul le remote-shell sera demandé respectivement sur les stations WS2, WS3 et WS4. Dans le même ordre d'idée, les fichiers contenant les résultats seront WSDEUX.DAT, WSTROIS.DAT et WSQUATRE.DAT. L'ordre identique des commandes dans les quatre modules a été voulu pour des raisons évidentes de facilité d'implémentation des modules en aval. Le code des programmes se trouve à l'annexes F.

module ELECTRE

La technique est également identique aux modules précédents. Le programme shell de ce module est accessible à l'annexe F. La différence supplémentaire réside ici dans la présence de deux nouvelles commandes que nécessite la présence de mesures des disques du serveur s'y rapportant. Il est à remarquer que ces commandes (*iostat -t* et *df*) ont été intercalées afin de maintenir la position terminale de la commande W.

module COMPWS1

Ce module de filtre est basé sur l'utilisation de l'outil de filtrage de fichier texte *awk* offert par le système Unix. Cet outil repose sur deux concepts essentiels : la ligne et le champ. Deux lignes sont séparées par un séparateur de ligne (retour chariot) visible à l'écran et des champs sont séparés par un simple caractère d'espacement (blanc). Ce système permet donc de scanner le fichier WSUN.DAT ligne par ligne et d'extraire le ou les champs qui nous intéressent.

Nous allons parcourir la logique de conception de ce module et la séquentialité des opérations de sélection de l'outil *awk* prévues pour garantir la réalisation des postconditions de ce module.

Les deux premières lignes du programme shell sont des instructions de contrôle qui nous permettent de vérifier que le programme a bien été activé par le démon *cron*.

La troisième ligne utilise l'instruction *cat* qui permet la concaténation des fichiers INIT1.DAT et WSUN.DAT dans un fichier temporaire AUXUN.DAT. La nécessité d'une telle opération repose sur la présence du fichier INIT1.DAT, ce dernier servant de compteur. Il permet en effet de conserver les valeurs des mesures cumulatives à la prise de mesure précédente (quinze minutes avant). Ce fichier contient douze valeurs correspondant dans l'ordre et sur une seule ligne aux mesures suivantes : nombre de paquets entrant dans l'hôte, nombre d'erreurs sur ces entrées, nombre de paquets sortant de l'hôte, nombre d'erreurs sur ces sorties, nombre de collisions, nombre de pages en entrées, nombre de pages en sortie, nombre de défauts de page, nombre de pages libérées, nombre de changements de contexte du processeur, nombre d'interruptions et nombre d'appels système. Ces données sont donc utilisées pour calculer la différence lors d'une nouvelle prise de mesures. Les deux instructions qui suivent dans le programme shell sont deux *awk*.

Le premier permet le filtrage des données du fichier AUXUN.DAT (résultat de la concaténation) vers le fichier COMPWS1.DAT. Dans la partie *BEGIN* du *awk* les variables sont initialisées à 0 pour permettre de respecter les postconditions au cas où le fichier WSUN.DAT serait vide. Vient ensuite dans le *awk* toute une série d'instruction de sélection. Cette partie du *awk*, contrairement au *BEGIN* et au *END*, est exécutée pour chaque ligne du fichier. Les sélecteurs permettent d'effectuer une série d'opérations lorsque le numéro de la ligne répond à la condition de sélection. Généralement les opérations effectuées sont des affectations d'un des champs de la ligne vers une variable locale au *awk*. Le premier sélecteur n'est activé que pour la première ligne du fichier qui n'est autre que les valeurs contenues dans le fichier INIT1.DAT, ces valeurs sont placées dans des variables locales que nous réutiliserons plus loin. Le deuxième sélecteur s'adresse aux lignes 4 à 13, si on regarde le contenu du fichier AUXUN.DAT en annexe H, on remarque que ces lignes correspondent aux 10 mesures instantanées de la commande *vmstat 1 10*. Les opérations effectuées sont des affectations cumulatives sur les valeurs qui nous intéressent. Les premières opérations sont particulières en ce sens qu'elles permettent la détection d'un caractère à la fin d'un nombre, en assure l'effacement et multiplie le nombre par le facteur que représente ce caractère. Le troisième et le quatrième sélecteur permettent de filtrer le temps moyen entre défaut de page et entre deux pages en entrée. Le cinquième sélecteur n'est activé que pour la ligne 20 et permet le calcul des mesures inhérentes à l'accès au réseau, comme il s'agit de cinq valeurs cumulatives, on va utiliser le contenu du fichier INIT1.DAT (annexe I), contenu qui a été affecté à plusieurs variables locales.

Si la nouvelle mesure est supérieure ou égale à l'ancienne alors le résultat sera la différence entre les deux, sinon (dans le cas d'une réinitialisation du système) le résultat sera la mesure qui vient d'être faite.

Cette technique est aussi appliquée pour les autres valeurs, dont le compteur est stocké dans le fichier d'initialisation, et qui sont affectées dans les sélecteurs des lignes 30, 31, 34, 46, 47, 48 et 51. Le dernier sélecteur permet quant à lui de filtrer la charge moyenne du système. Vu que le format de la ligne est variable, nous sommes obligé de stipuler au filtre de prendre les trois dernier champs de la ligne ce qui se traduit par le nombre de champs de la ligne (NF) duquel on retire 0, 1 ou 2.

La partie *END* renferme deux types d'instructions. Tout d'abord les opérations de calcul de moyenne pour les valeurs provenant de la commande *vmstat 1 10* et ensuite les opérations d'impressions dans le fichier de destination selon le format défini par la séquence *s%\t* pour signifier une séparation par une tabulation et *s%\n* pour demander un passage à la ligne pour la valeur suivante.

Le second *awk* permet de remettre dans le fichier INIT1.DAT les nouvelles mesures. Le format de ce fichier esaccessible en annexe I. Cette mise-à-jour du fichier n'a lieu que si le fichier WSUN.DAT n'est pas vide afin d'éviter une erreur des mesures. Effectivement le filtre *awk* ne trouverait dans ce cas aucune ligne dans le fichier WSUN.DAT et imprimerait dans le fichier INIT1.DAT les valeurs initialisées à 0, ce qui aurait pour conséquence de rendre les mesures incorrectes.

module COMPWS2,COMPWS3 et COMPWS4

L'implémentation de ces trois modules est totalement identique au module COMPWS1. Seuls les noms de fichiers doivent être adaptés, ainsi pour le module COMPWS2, WSUN.DAT devient WSDEUX.DAT, INIT1.DAT devient INIT2.DAT (annexe I), AUXUN.DAT devient

AUXDEUX.DAT (annexe H) et COMPWS1.DAT devient COMPWS2.DAT. La même logique est appliquée pour les deux autres modules.

module COMPELECTRE

Les remarques faites pour les trois modules précédents sont d'application au module COMPELECTRE. Toutefois une différence supplémentaire justifie sa position séparée des autres. Des mesures concernant l'activité des disques devant être faites, le format du fichier ELECTRE.DAT est différent des fichiers précédents, du moins à partir de la ligne 19. Il découle de cette remarque que les sélecteur de ligne sont différents et surtout que sept nouveaux sélecteurs ont été ajoutés pour permettre le filtrage des mesures concernant les disques. Ces sélecteurs sont ceux qui correspondent aux lignes 21 à 30, 33, 34, 35, 36, 37 et 40. Le reste de l'implémentation est semblable aux modules précédent si ce n'est le format d'affichage qui a été complété dans le *END* du premier *awk*.

module COMPHEURE

Le module COMPHEURE est également basé sur l'utilisation du *awk* pour calculer la moyenne horaire et donner un résultat respectant les postconditions. Nous allons décrire de la même manière que pour les programme précédents, l'implémentation de ce module.

Les deux premières lignes du programme shell sont à nouveau des commandes de contrôle de bon exécution.

La troisième ligne est importante car elle concatène les cinq fichiers COMP*.DAT DANS fichier temporaire AUX.DAT qui sera utilisé en entrée du premier *awk*.

Le premier *awk* assure la sélection des champs des vingt groupes que contient le fichier AUX.DAT. Nous n'allons pas revoir en détails tous les sélecteurs, nous en expliquerons un de façon approfondie. Le cinquième sélecteur est actif pour les lignes 53 ou 57 ou 61 ou 65. Ces numéros de lignes correspondent à la première ligne des quatre groupes provenant du fichier COMPWS3.DAT et se rapportant aux mesures sur le CPU.

Les opérations effectuées si le sélecteur est activé sont des affectations cumulatives dont le résultat sera réutilisé pour calculer la moyenne horaire. Le filtre `awk` travaille en entrée sur le fichier `AUX.DAT` résultat de la concaténation et imprime ses résultats dans le fichier temporaire `HEURE.DAT` (annexe K). Nous avons été obligé de séparer le travail de filtrage en deux `awk`, le nombre d'opérations de sélection étant trop grand que pour les regrouper dans un seul. Le deuxième `awk` n'est donc rien d'autre que la seconde partie du filtrage et du calcul de moyenne. Les valeurs citées dans l'implémentation du module `COMPWS1` et comptabilisées au moyen des fichiers `INIT*.DAT` ne sont pas ramenées à une moyenne horaire dans la partie *END* du `awk` contrairement aux autres mesures.

La commande de concaténation fusionne le fichier `HEURE.DAT` avec le fichier `PERFORMANCE.JC` dans un fichier auxiliaire `AUX2.DAT`.

Le dernier `awk` prend les résultats des deux précédents stockés dans le fichier `AUX2.DAT` et les écrits selon le format prévu dans le fichier `PERFORMANCE.JC`. Les cinq premières lignes de `AUX2.DAT` correspondent aux cinq domaines de mesures et sont déjà dans le format du fichier de destination `PERFORMANCE.JC`. L'algorithme contenu dans le `awk` consiste donc à gérer la position des lignes dans le fichier.

module COMPJOUR

L'implémentation de ce module est similaire dans sa logique au programme `COMPHEURE`. Le but étant de sélectionner les valeurs de chaque heure et ce par mesure et d'en calculer la moyenne. La moyenne de toutes les lignes du fichier `PERFORMANCE.JC` constitue donc une ligne du fichier `PERFORMANCE.MC`.

La troisième ligne du module copie le contenu du fichier `PERFORMANCE.JC` dans le fichier `PERFORMANCE.JP` qui correspond aux mesures de la veille.

Le troisième *awk* contient un algorithme identique à celui repris dans le module COMPHEURE gérant la position des lignes dans le fichier PERFORMANCE.MC.

La dernière instruction permet de réinitialiser le fichier PERFORMANCE.JC au vide.

module COMPMOIS

Outre les deux première commandes qui sont des instructions de contrôle d'exécution, ce module assure l'archivage des mois écoulés.

La troisième ligne copie le contenu du fichier du mois courant dans le fichier du mois précédent. La ligne suivante stocke dans un fichier temporaire la date courante. Ce dernier fichier est concaténé avec le fichier PERFORMANCE.MC et le fichier ARCHIVE dans un second fichier provisoire AUX3.DAT.

Le *awk* a pour but remettre à jour le fichier archive en y insérant le mois en question suivi du contenu du fichier PERFORMANCE.MC.

La dernière opération réinitialise le fichier PERFORMANCE.MC au vide.

module COMPBACKUP

En plus des lignes de contrôle, ce module permet au moyen d'un fichier provisoire AUXB.DAT, la mise à jour du fichier PERFORMANCE.JC en y ajoutant des lignes contenant des valeurs toutes nulles et correspondant aux prises de mesures rendues impossibles par les opérations de backup du lundi matin.

(4) Conduite des test individuels

Comme nous l'avons prévu lors de l'élaboration des plans de tests, le seul test réalisable avec les modules tels qu'ils ont été développés est une exécution, parfois combinée avec un contexte argumentera variable.

Vu la séquentialité que nécessite l'exécution de ce logiciel, les test individuels successifs ont déjà joué le rôle partiel des test d'intégration que nous développerons plus avant dans ce chapitre.

Nous avons rencontrer, lors de la réalisation de ces test, deux types d'erreurs. Les erreurs syntaxiques et les erreurs sémantiques. Les erreurs syntaxiques peuvent être définie comme les erreurs détectées par l'interpréteur de commande et particulièrement dans l'outil de filtrage *awk*. Les erreurs sémantiques peuvent quant à elles être définies comme les erreurs portant sur la cohérence et/ou la correction des valeurs filtrées. Les erreurs syntaxiques inhérentes à la commande *awk* ont été détectées immédiatement par l'interpréteur de commande, cette détection faisant obstacle à l'exécution du module. Les erreurs sémantiques telles que nous les avons définies ne peuvent être détectées que par vérification manuelle des valeurs filtrées.

(a) Erreurs syntaxiques

Deux catégories particulières d'erreurs syntaxiques ont été rencontrés lors de la conduite de ces tests. Tout d'abord les erreurs de syntaxe pure du langage de programmation shell (utilisation d'un mot réservé comme nom de variable, ...) ou du langage de sélection de la commande *awk* (double signe = dans une condition de sélection, ...). Ces erreurs sont détectées assez facilement, malgré le manque de message d'erreur précis du système, à l'aide du numéro de la ligne fautive renvoyé par l'interpréteur.

Ensuite les erreurs que l'on pourrait qualifier de saturation du *awk* provenant d'un excès d'instruction de sélection à l'intérieur d'un seul *awk*. Ce type d'erreur nous a obligé à séparer en deux parties le *awk* contenu dans le module COMPHEURE.

(b) Erreurs sémantiques

Les erreurs sémantiques sont typiquement basées sur une erreur de sélection du champs ou de la ligne. Ce type d'erreur a pour conséquence de ne pas faire obstacle au filtrage mais d'occasionner une erreur dans la valeur attendue. Cette valeur peut être non numérique ou numérique mais incorrecte, si la détection de la première est assez immédiate, la seconde nécessite une vérification de chaque valeur à partir du fichier de départ.

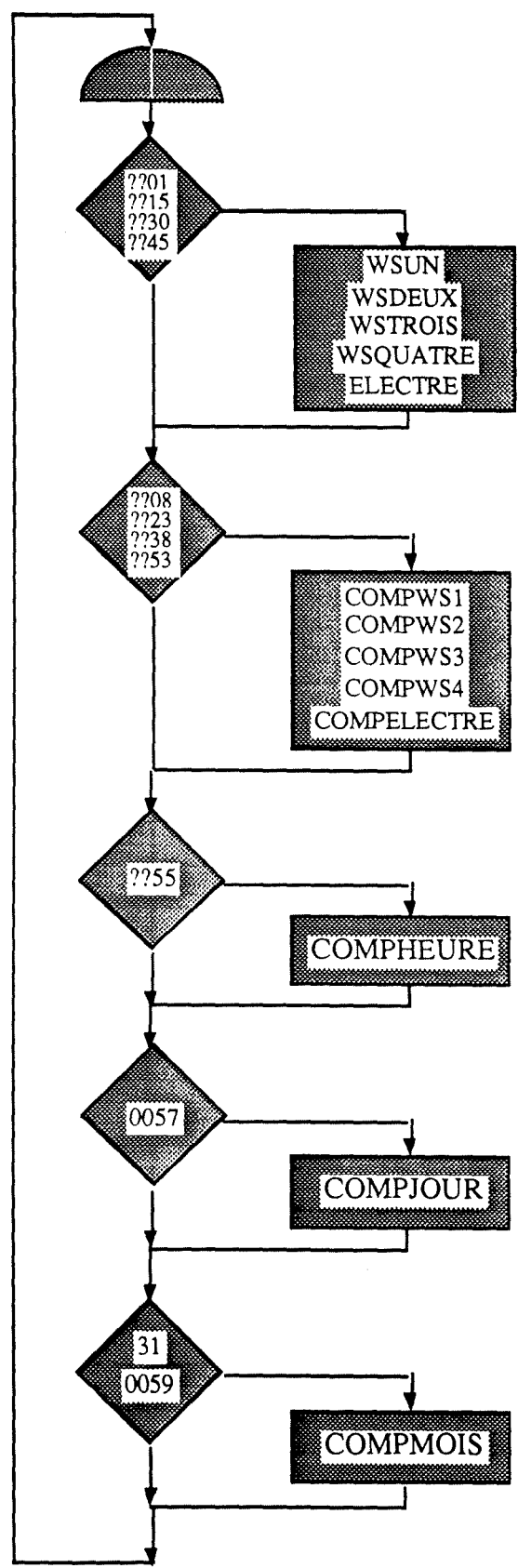
Les valeurs numériques incorrectes peuvent provenir d'une erreur de sélection du champs mais aussi d'une erreur de calcul d'une moyenne, notamment dans les module COMP*.

(c) Intégration

Au cours de cette phase nous allons aborder le problème de la jonction de tous les modules développés distinctement et les tests y afférant. La mise en place et le démarrage de l'outil feront partie de cette étape d'intégration car la mise en oeuvre du logiciel représente une partie des tests d'intégration.

(1) Conception du coordinateur

Les modules ayant été testés individuellement, il convient de les placer dans un contexte réel d'exécution. Pour ce faire, il faut mettre au point la coordination de leurs exécutions respectives. Le coordinateur aura donc un rôle de synchronisation et ne sera nullement interactif, cette synchronisation sera basée sur l'algorithme suivant :



Il faut ajouter à cet algorithme schématique le module **COMPBACKUP** ainsi que les déphasements dont il est l'origine. Effectivement tous les lundis entre 0800 Hr et 1200 Hr cet algorithme est déconnecté et remplacé par le module précité.

L'exécution de cet algorithme de synchronisation est assurée par une table du système, appelée *crontab*, gérant l'activation des processus dont les moments de déclenchements doivent être programmés. La partie du contenu de cette table se rapportant à notre module coordinateur se trouve à l'annexe L.

Les critères temporels de déclenchements sont donnés par les cinq premiers champs de la ligne, chaque ligne se rapportant à un programme particulier. Le premier champs stipule les minutes [0-59], le deuxième les heures [0-23], le troisième le jour du mois [1-31], le quatrième le mois de l'année [1-12] et le cinquième le jour de la semaine [1-7]. Le dernier champs renseigne le programme à exécuter avec le chemin complet pour l'atteindre.

Cette *crontab* est scannée chaque minute par un **processus démon** du système qui y sélectionne les programmes à exécuter.

On peut donc décrire en détail les conditions d'exécutions de chaque module et vérifier qu'elles correspondent bien à l'algorithme.

(2) Tests d'intégration et installation de l'outil

Nous avons regroupé en un seul point deux étapes qui a priori sont totalement différentes car en réalité elles sont très proches et l'installation de l'outil constitue une partie des tests d'intégrations.

(a) Exécution symbolique

L'exécution symbolique consiste en une exécution des différents modules dans la séquence prévue par l'algorithme de synchronisation et de telle sorte que les préconditions et postconditions de chaque module soient respectées. On enlève donc de l'algorithme de coordination les contraintes temporelles ce qui permet une exécution et donc un contrôle plus rapide et aussi rigoureux conduisant à la détection des erreurs comme une exécution réelle.

Ce premier test d'intégration nous a permis de détecter une erreur assez typique. Les fichiers COMPWS*.DAT et le fichier COMPELECTRE.DAT n'étaient jamais effacés, ce qui a pour conséquence de cumuler les unes à la suite des autres toutes les valeurs filtrées. Lors du calcul de la moyenne horaire par le module COMPHEURE, aucune erreur n'est détectée mais la moyenne est constamment calculée sur les mêmes valeurs (les 4 premiers groupes). Nous avons donc modifié le module COMPHEURE pour qu'il efface les fichiers COMPWS*.DAT et COMPELECTRE.DAT après chacune de ses exécutions.

(b) Démarrage du système

Ce second test d'intégration introduit la notion de contrainte temporelle par rapport à l'exécution symbolique. Cet apport nous permet de détecter non plus des erreurs de conception mais des erreurs d'exécution. L'erreur d'exécution peut être dans ce contexte considérée comme une erreur provenant de l'environnement technique du logiciel et doit être résolue dans ce contexte. Cette mise en service de l'outil nous a mis face à un problème technique en rapport avec la procédure de remote-shell déjà évoquée plus avant.

L'accès aux stations à partir du serveur Electre était refusé par le système vu que les processus définis dans la *crontab* et exécutés par le démon *cron* appartiennent à *root* et que ces processus doivent accéder à des données appartenant à *dedo*. Le fait de spécifier le nom d'utilisateur *dedo* dans la définition des processus de la *crontab* ne suffisait pas à obtenir l'autorisation d'accès en raison de l'utilisation par le système d'une série de paramètres d'environnement durant l'activation des processus. Ces paramètres n'étaient en effet pas initialisés dans le *shell*, il fallait donc les initialiser et les exporter. C'est exactement ce que font les premières lignes des modules de prise de mesures après avoir fixé une protection par défaut des fichiers. La commande *\$shell* utilise le *shell* défini comme variable d'environnement et les commandes de mesure en arguments.

3.3.3.2.1 Module de présentation

Si nous avons suivi une méthodologie de développement assez classique pour la partie du logiciel responsable de la collecte des données , il en est tout autrement pour le module de présentation des résultats obtenus.

Ce module est réellement très particulier en ce sens que nous avons voulu qu'il réponde à certains critères déjà évoqués dans le cahier des charges. Nous avons déjà expliqué dans les choix de départ l'optique que nous avons prise pour mettre au point cet outil particulier, nous allons y revenir en détails tout au long de ce paragraphe. Nous aborderons ensuite la logique de développement et nous terminerons par décrire en détails l'implémentation du module.

(a) Choix de l'environnement

Notre choix s'est porté, comme nous l'avons déjà suggéré, sur le logiciel d'aide à la décision Excel de Microsoft sur Macintosh. Sa caractéristique de système informatique d'aide à la décision rencontre parfaitement nos aspirations. En effet, que pourrait-on trouver de plus approprié qu'un tableur pour présenter des ensembles de mesures évolutives.

Nous allons donner de façon détaillée les arguments qui nous ont mené à ce choix. Ces critères ayant déjà été définis dans le point consacré aux choix de départ, nous nous bornerons ici à les aborder d'un point de vue plus pratique, plus proche de la réalité des besoins ressentis par l'administrateur.

Premièrement ce environnement, tant logiciel que matériel, est non seulement bien connu mais aussi très implanté au sein de l'Institut d'Informatique. Son utilisation classique est donc acquise pour bon nombre de personnes. Même si la technique des commandes groupées en macro-commande est moins utilisée, un développement judicieux de ces dernières peut en masquer le contenu et tout en personnalisant les fonctionnalités du tableur, on maintient l'environnement original d'Excel. Cette technique a pour effet d'augmenter la simplicité au sens où nous l'avons déjà définie.

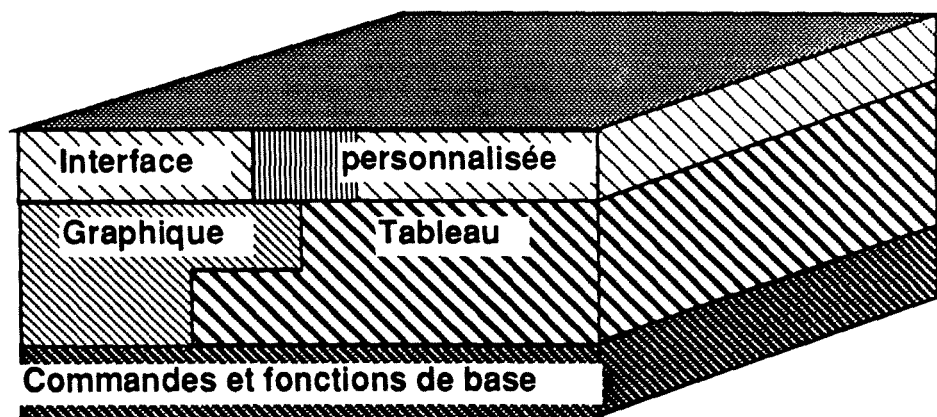
Deuxièmement, la convivialité légendaire du Macintosh en général et du logiciel Excel en particulier, est un argument de poids dans le choix de cet environnement. La convivialité est dans ce cas une garantie d'utilisation rapide et plus fréquente de notre outil. Cette utilisation plus fréquente peut être également accrue par le fait que l'exécution de l'outil de présentation dans cet environnement peut se faire en partage avec d'autres applications Excel.

Troisièmement, la disponibilité d'un outil graphique évolué répondant exactement au cahier des charges représente un avantage considérable. La variété des présentations graphiques qu'offre Excel couvre largement les besoins répertoriés.

Enfin, la réutilisabilité des représentations sélectionnées peut être considérée comme maximale. Les possibilités de réutilisabilité dans l'environnement sont triples. Tout d'abord la possibilité de sauver sur mémoire de masse (disque dur ou disquette) le résultat obtenu, ensuite l'impression indépendante de ce résultat sur papier et enfin le transfert de cette sélection vers une application différente, généralement la rédaction d'un rapport sous un logiciel de traitement de texte ou de publication (Word, Macwrite, Pagemaker, ...). Cette réutilisabilité nous est apparue comme essentielle pour une application de gestion, domaine dans lequel l'information n'est pas faite pour être oubliée mais bien pour faire l'objet d'une diffusion voir d'une interprétation.

(b) Logique de développement

La logique de développement de ce module de présentation repose sur une découpe en couches assez classique garantissant une excellente lisibilité du logiciel et de ce fait sa maintenabilité. La découpe en niveau peut être schématisée de la manière suivante :



Cette découpe montre que la présentation en tableau est la base de l'application. L'affichage graphique se sert de l'affichage en tableau. Vient se superposer à ces deux couches imbriquées la partie interface utilisateur gérant les menus, commandes et autres zones de dialogue. Cette couche se décompose en 3 parties de même niveau, une partie propre à chaque type de présentation et une partie commune. A la base

de cette découpe nous trouvons la couche réunissant les commandes et les fonctions standards du logiciel Excel.

Le principe de base de cette application étant brossé, nous allons décortiquer en détails l'implémentation de l'outil de présentation.

(c) Implémentation

Pour maintenir une certaine traçabilité avec la logique de développement, nous baserons notre raisonnement sur les deux fonctionnalités principales du logiciel représentées par la couche intermédiaire imbriquée du schéma de la découpe. Nous rapporterons tout au long de cette partie au texte de la **macro-commande** d'Excel repris en annexe M. La première de ces deux fonctionnalités vise à donner à l'administrateur du système de présenter sous forme de tableau les données qu'il a sélectionnées.

Les 280 premières lignes de la **macro** définissent les menus, zones de dialogue et commandes nécessaires à l'utilisateur pour d'une part sélectionner les données et d'autre part demander l'exécution de sa sélection.

Nous avons utilisé le maximum des possibilités offertes par Excel pour accroître la simplicité et la convivialité de l'application. Tout d'abord une barre de menu personnalisée donne accès aux commandes parfois suppléée par une zone de dialogue. Trois types de menu peuvent être distingués, le menu FICHER rappelant les commandes classiques d'Excel et qui constitue la partie commune de la couche interface, les menus de sélection PERIODE, DONNEES et HOTES et enfin le menu d'activation de l'application AFFICHAGE.

Les messages d'erreurs ont été prévus pour tous les cas détectés et ce de façon claire et explicite dans le plus pur style propre au Macintosh.

La possibilité est également offerte à l'utilisateur d'employer des "**raccourcis-clavier**" bien connus des usagers, cette possibilité permet une augmentation de la vitesse d'utilisation en diminuant l'interaction entre l'application et l'utilisateur. Elle s'adresse à l'utilisateur quelque peu habitué avec les commandes du logiciel.

Une aide directe est également fournie à l'administrateur chaque fois qu'il sélectionne une commande dans un menu, cette aide est affichée dans la barre de dialogue au bas de l'écran. Outre ces caractéristiques d'interfaçage, cette partie du code contient les macro-commandes gérant les choix du menu fichier.

Les lignes 22 et 23 contiennent les paramètres déterminés par la macro-commande construisant la table sur base des critères de sélection introduits par l'utilisateur.

A partir de la ligne 282 débute la macro-commande qui gère la construction et la présentation du tableau. Nous allons la parcourir sans entrer dans les détails de l'implémentation mais en esquissant les actions essentielles qu'elle génère.

Les lignes 283 à 285 teste si une présélection a bien été faite sur les critères disponibles (période, données et hôte). La macro-commande teste ensuite la période qui a été choisie et ouvre le fichier correspondant (ligne 289 à 290). Les lignes 294 à 298 permettent le remplacement, si nécessaire, des séparateurs décimaux "." (sur les hôtes Unix en ",", sur Excel. La macro-commande teste au moyen d'un **drapeau** si cette modification est nécessaire. Si un changement a été opéré, le fichier est sauvegardé, afin d'éviter de devoir le modifier de nouveau lors d'une éventuelle autre utilisation ; le drapeau est également modifié.

Les lignes 304 à 523 renferment toutes les procédures de test qui vont permettre de placer dans les cellules de la ligne 22 les variables nécessaires (zone de la feuille de calcul contenant les données brutes) à la construction du tableau sur base des critères présélectionnés.

Les lignes 524 à 777 construisent le tableau selon la séquence suivante :

- Inscription du titre du tableau et mise en forme.
- Inscription et mise en forme des titres de colonnes.
- Encadrement des titres de colonnes.
- Inscription et mise en forme de l'échelle (jour ou heures).
- Encadrement de la colonne contenant l'échelle.
- Activation de la feuille de données brutes.
- Sélection de la zone d'intérêt.
- Copie
- Activation de la feuille de calcul qui contiendra le futur tableau.
- Sélection de la zone de destination.
- Collage
- Calcul de la moyenne et insertion en bas de colonne.
- Encadrement des colonnes contenant les résultats.

A partir de la ligne 779 débute la macro-commande d'affichage graphique. La première commande exécutée étant l'enregistrement des paramètres d'affichage graphique. Vient ensuite l'exécution de la macro-commande gérant la construction du tableau sur lequel se base le graphique. Cette exécution est toutefois soumise à un test, elle ne sera pas exécutée si la construction du tableau a déjà été réalisée et que les critères de sélection n'ont pas été modifiés.

Les lignes 783 à 786 modifie l'interface utilisateur, ce qui correspond à la découpe en couche décrite au point précédent.

La macro-commande sélectionne ensuite la zone des résultats sur la feuille de calcul contenant le tableau. Cette zone est déterminée par un nombre de colonnes (fonction du domaine choisi) et par un nombre de lignes (fonction de la période sélectionnée).

Les lignes 797 à 816 dessinent le graphique et le mettent en forme selon des critères en partie fixés (format des axes) et des choix fait par l'administrateur (type de graphique, légende).

Les lignes 818 à 862 regroupent les commandes accessibles via les menus offerts en mode graphique à l'utilisateur, tels que modifier le type de graphique, sauver le graphique sur disque, relancer la construction du graphique

avec les nouveaux paramètres ou encore quitter le mode graphique ou ajouter une flèche au graphique.

Les lignes 864 à 896 constituent la macro-commande qui génère un graphique en secteurs basé sur les moyennes calculées au pied du tableau.

(d) Erreurs rencontrées

Les erreurs les plus souvent rencontrées dans la mise au point de cette application ont été de deux types que l'on pourrait qualifier de similaires à ceux rencontrés lors du développement du module de prise de mesures. Les erreurs de syntaxes très typique au langage des macro-commandes Excel. Ce langage est non-structuré, ce qui oblige le "programmeur" à manoeuvrer de manière peu algorithmique. Les nombreuses fonctions *atteindre()* en témoignent d'ailleurs à souhait. Outre ces erreurs de syntaxe, sont également apparues des erreurs d'exécution pure généralement faciles à détecter et à corriger.

Le principe même de la macro-commande basée sur un langage non-structuré et ne reposant sur aucun algorithme nous impose d'émettre des réserves quant à la preuve de sa correction. Aucune preuve de correction ne peut être apportée et les tests effectués ne sont pas exhaustifs. Un maximum de combinaisons ont été testées mais cela ne garantit en rien l'absence d'erreur.

Le type d'application permet toutefois une souplesse importante palliant ce manque de garantie de correction. D'une part il est aisé de retourner au menu général d'Excel ou de sortir de l'application et de la relancer assez rapidement, et d'autre part, aucune donnée n'est modifiée ni perdue, les risques d'incohérence sont donc nuls et il est toujours possible de récupérer rapidement les résultats désirés. L'absence de modifications des données brutes permet donc de garantir la correction des résultats sans garantir l'absence d'erreurs.

3.4. UTILISATION DU LOGICIEL

3.4.1. INTRODUCTION

Le but de ce point est de permettre un apprentissage rapide et un usage efficace du moniteur développé. Il constitue donc un véritable manuel d'utilisation décrivant les fonctionnalités du logiciel à travers un scénario d'utilisation.

Comme nous l'avons signalé à plusieurs reprises, le module de prise de mesures n'est pas interactif : il réalise sa tâche automatiquement , sans intervention humaine. Puisqu'il n'y a aucun "contact" entre l'utilisateur et le module de prise de mesures du moniteur, il sera ignoré dans ce point.

Le logiciel Kermit assurant le transfert des fichiers de données (du réseau vers Macintosh) ne sera pas abordé non plus. En effet, il peut être considéré comme connu vu son implantation au sein de l'Institut. De plus, son utilisation, pour assurer cette fonction de transfert, doit être considérée comme une suggestion, tout autre logiciel garantissant le même résultat pouvant être utilisé.

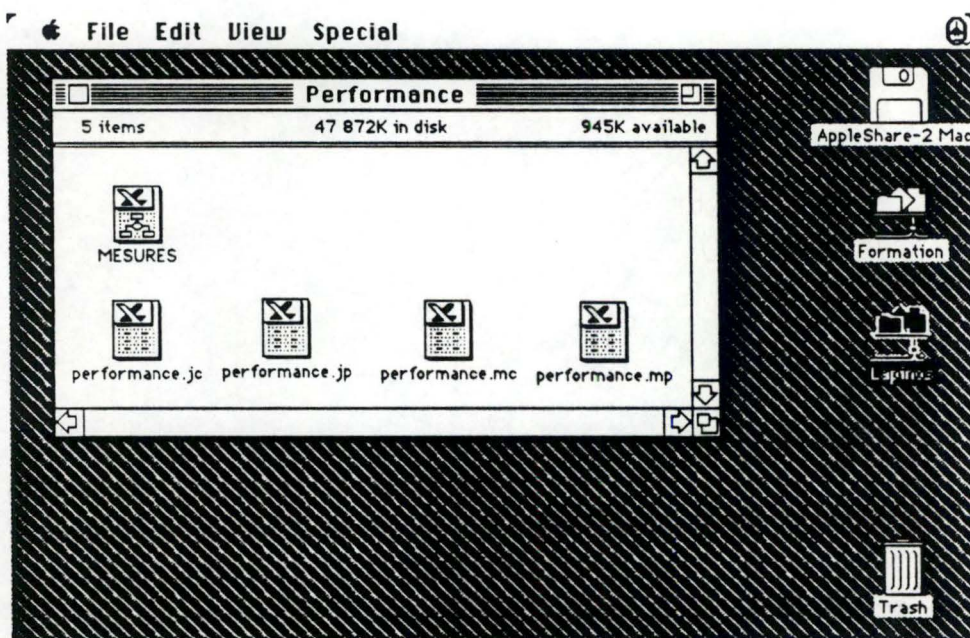
Nous aborderons donc exclusivement le module de présentation selon une approche que nous qualifierons de pédagogique, au sens où elle va permettre une découverte progressive de l'outil et de ses fonctionnalités. Cette démarche sera abondamment illustrée par les écrans générés aux différents moments de l'utilisation du logiciel.

3.4.2. MANUEL D'UTILISATION

L'utilisation du logiciel de présentation comporte trois grandes étapes successives : le lancement, la sélection des données et la construction d'un tableau et/ou d'un graphique.

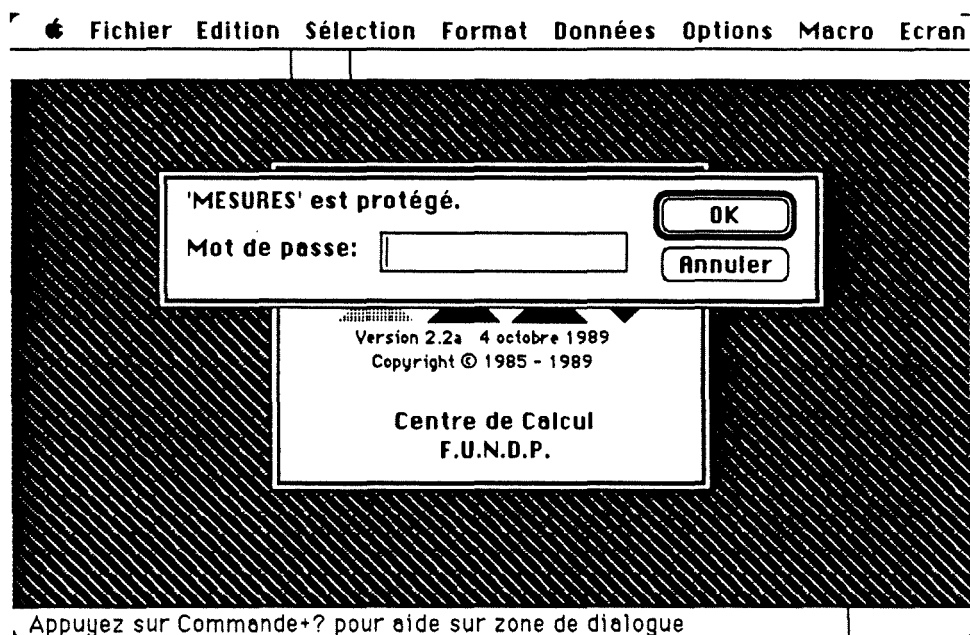
3.4.2.1. LANCEMENT DU LOGICIEL

Le lancement du module de présentation s'opère à partir de n'importe quel dossier sur disque dur ou sur disquette, l'utilisation d'un disque dur s'avérant naturellement plus performante.



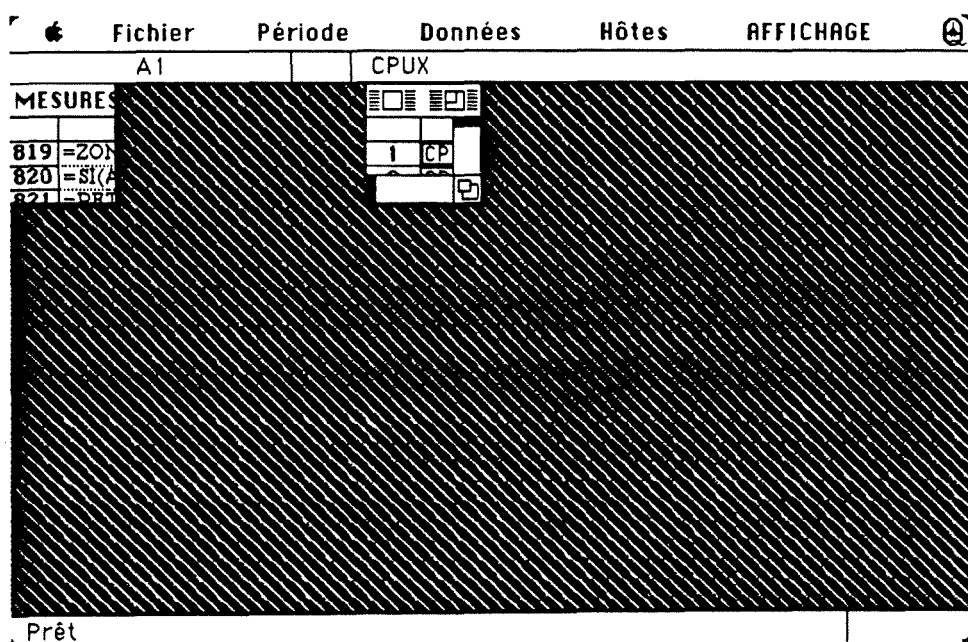
Comme l'indique cet écran, le module porte le nom MESURES et est iconifié comme une macro d'Excel. Les fichiers de données doivent se trouver dans le même répertoire que celui-ci. Le fichier PERFORMANCE.MC doit absolument y être présent puisqu'il est chargé par défaut au lancement (les autres étant facultatifs). Il faut également que ces fichiers gardent leur nom d'origine pour pouvoir être reconnu par le logiciel, à savoir PERFORMANCE.JC, PERFORMANCE.JP, PERFORMANCE.MC et PERFORMANCE.MP.

Le lancement de l'application s'opère par un double clic de la souris sur l'icône de MESURES. Ceci a pour conséquence l'exécution du logiciel développé sur Excel, celui-ci est toutefois protégé par un mot de passe qui peut être modifié (le mot de passe original étant "MP").



Une fois le mot de passe introduit, le programme macro s'autoexécute : il affiche la nouvelle barre de menu et charge le fichier de données brut PERFORMANCE.MC. Si ce dernier est ouvert pour la première fois, les séparateurs décimaux "." seront remplacés par des ",", et le fichier modifié sera enregistré, mais l'utilisateur n'a pas à s'en soucier.

L'utilisateur se trouve alors devant l'écran suivant, le logiciel n'attendant plus que ses ordres.

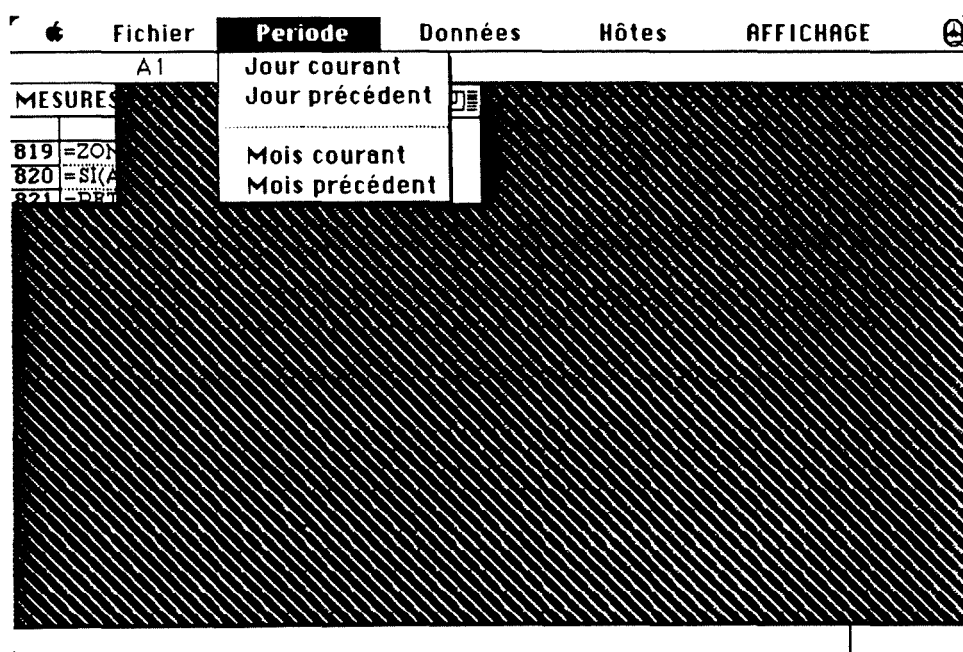


3.4.2.2. SELECTION DES DONNEES

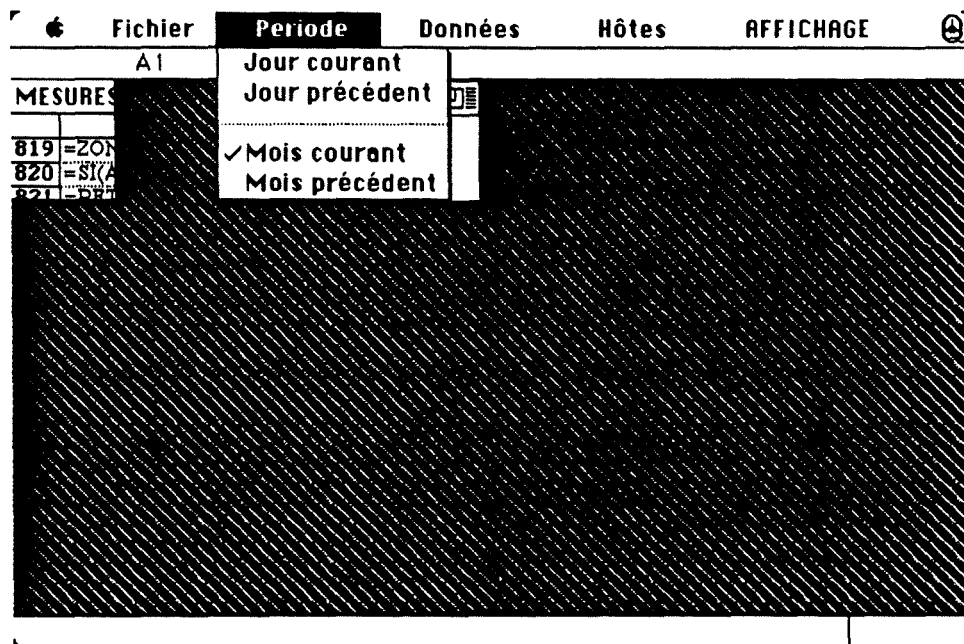
Le choix des données sur lesquelles se base la construction du tableau ou du graphique désiré, s'opère à l'aide de trois menus.

3.4.2.2.1. Sélection de la période

Quatre périodes sont disponibles et correspondent aux quatre fichiers créés par le module de prise de mesures. A ce niveau, ceux-ci doivent évidemment avoir déjà été transférés, la sélection d'une période non-présente provoquerait une erreur d'exécution. Le choix de la période se fait à l'aide du second menu déroulant.

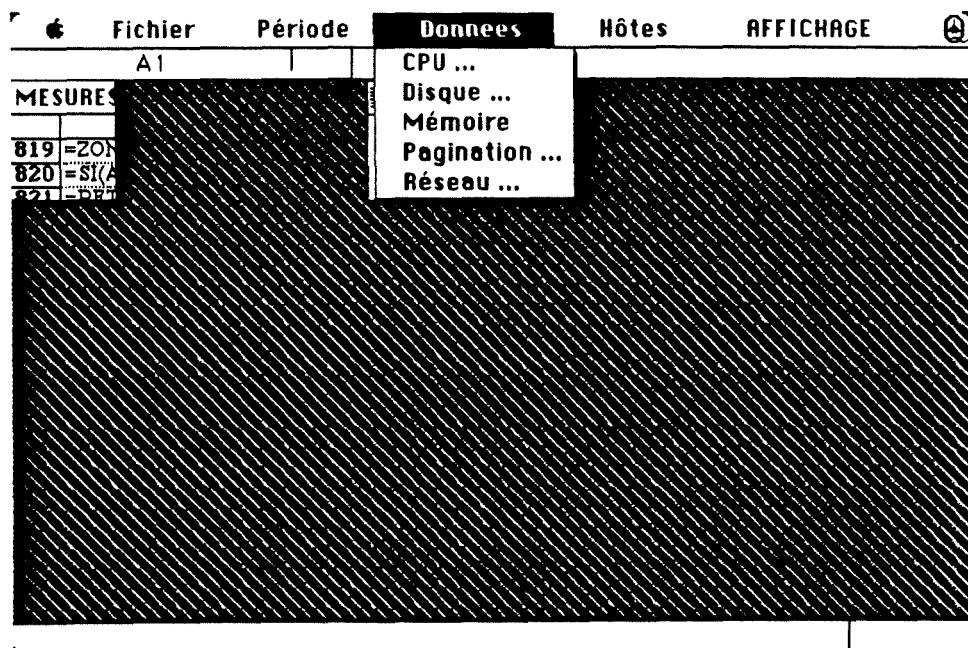


On pourra aisément vérifier la période sélectionnée en réactivant le menu déroulant, la coche en regard indiquant le choix effectué.



3.4.2.2.2. Sélection des données

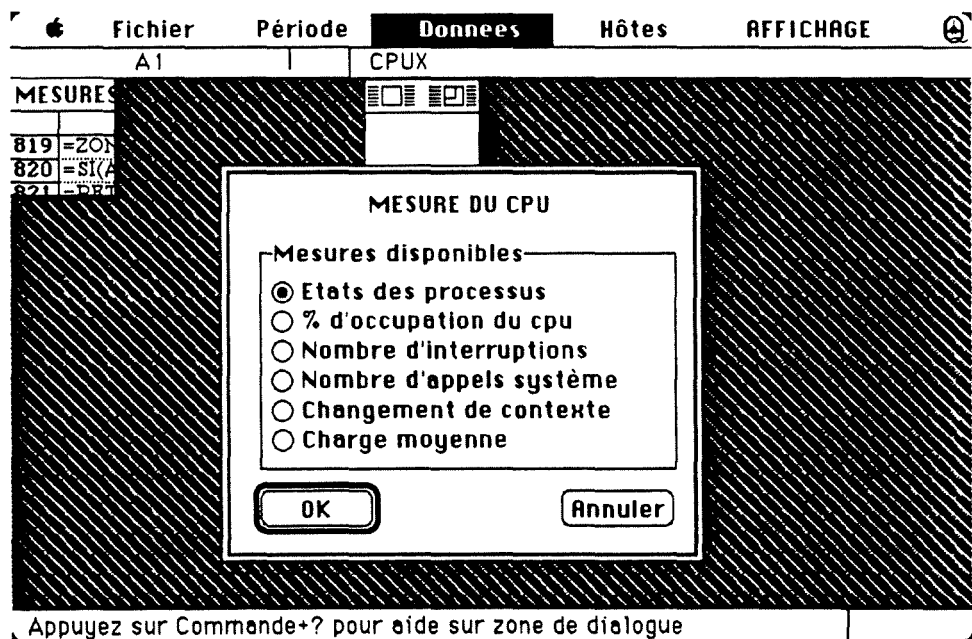
Après avoir choisi une période de référence, l'utilisateur doit sélectionner le domaine particulier de mesure qui l'intéresse. Les différents domaines sont accessibles via le troisième menu.



Ce menu offre le choix entre les cinq grandes classes de données qui sont l'objet d'une surveillance par le module de prise de mesures :

(a) CPU

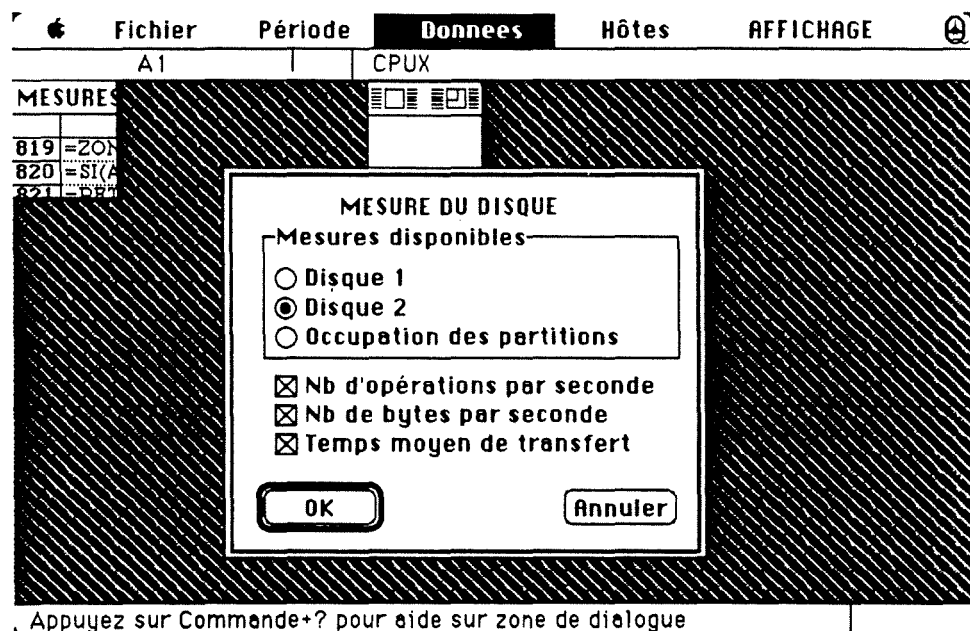
Le fait d'activer la sélection CPU entraîne (comme les trois points l'indiquent) l'affichage d'une zone de dialogue permettant à l'utilisateur d'affiner son choix.



Cinq types de données CPU sont disponibles via des boutons, les choix étant exclusifs. La sélection sera validée en cliquant le bouton OK, alors que cliquer le bouton Annuler aura pour conséquence de maintenir les sélections précédentes.

(b) Disque

Le choix doit être fait entre les mesures concernant les partitions et les différents disques éventuels. Il est possibles d'obtenir, sur le même graphique ou tableau, les mesures du nombre d'opérations effectuées par seconde, le nombre de bytes transférés par seconde et le temps moyen de transfert. Cette sélection additionnelle se fait grâce aux cases à cocher.

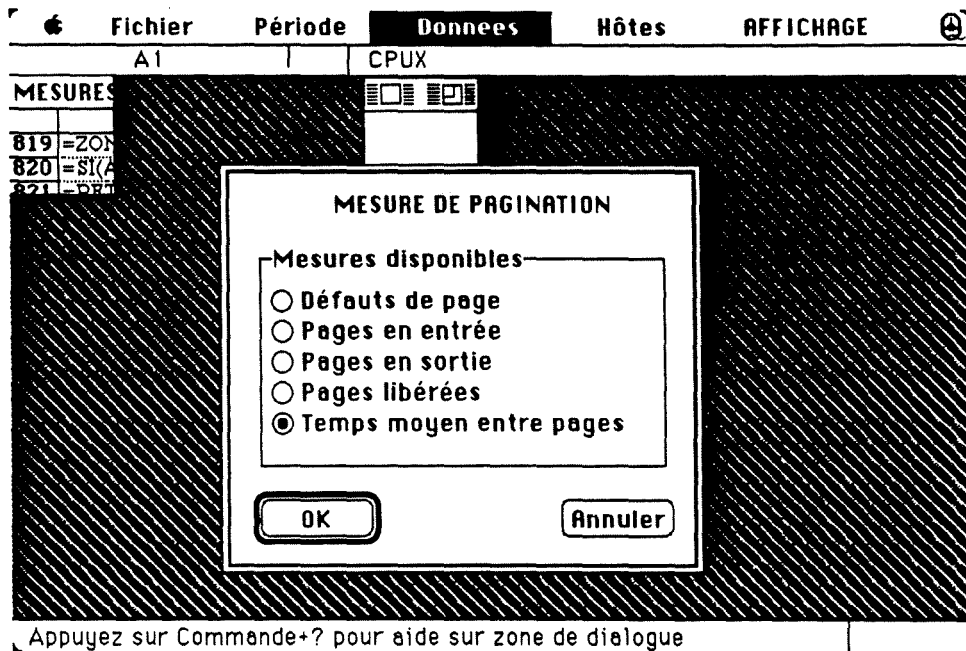


(c) Mémoire

Les mesures sur l'occupation de la mémoire sont obligatoirement la taille de la mémoire libre et de la mémoire virtuelle utilisée. Puisqu'il n'y a pas d'autres mesures possibles, aucune zone de dialogue n'apparaît, mais le choix Mémoire est bien effectué.

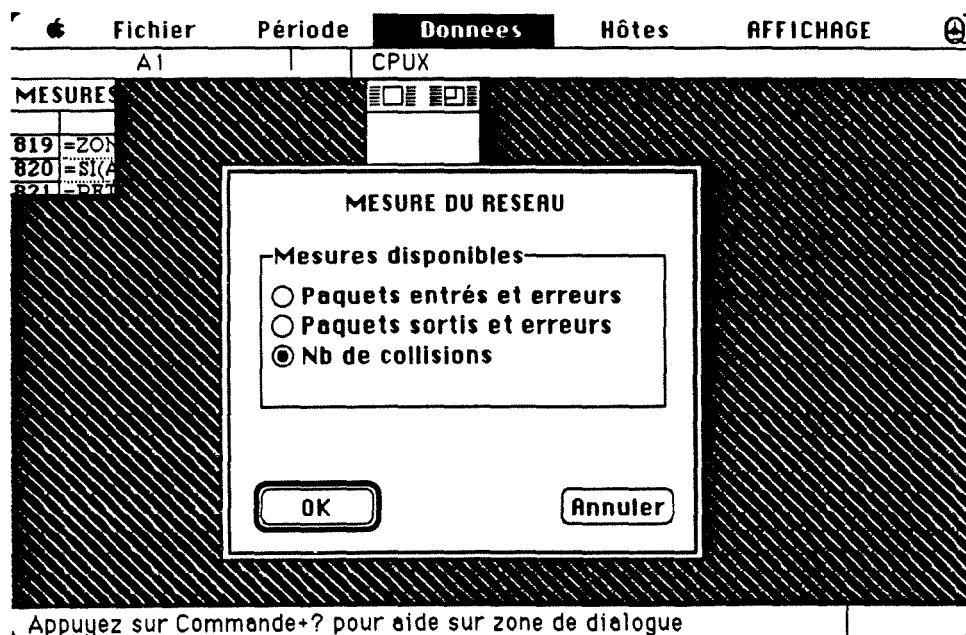
(d) Pagination

L'activité de pagination comprend cinq types de données dont les choix sont mutuellement exclusifs.



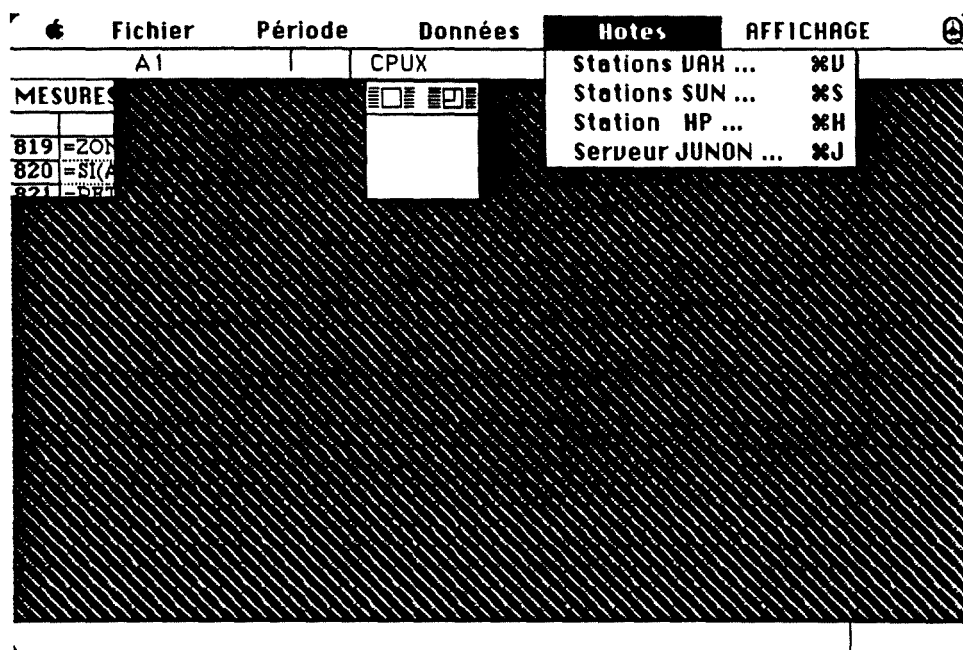
(e) Réseau

L'utilisateur doit porter son choix sur une des trois données concernant les accès au réseau.

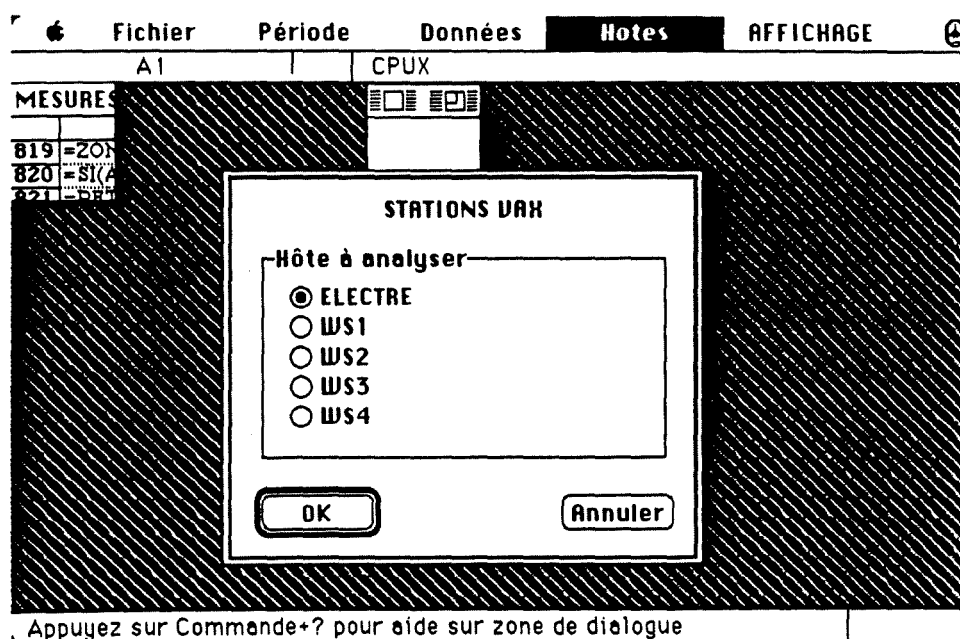


3.4.2.2.3. Sélection de l'hôte

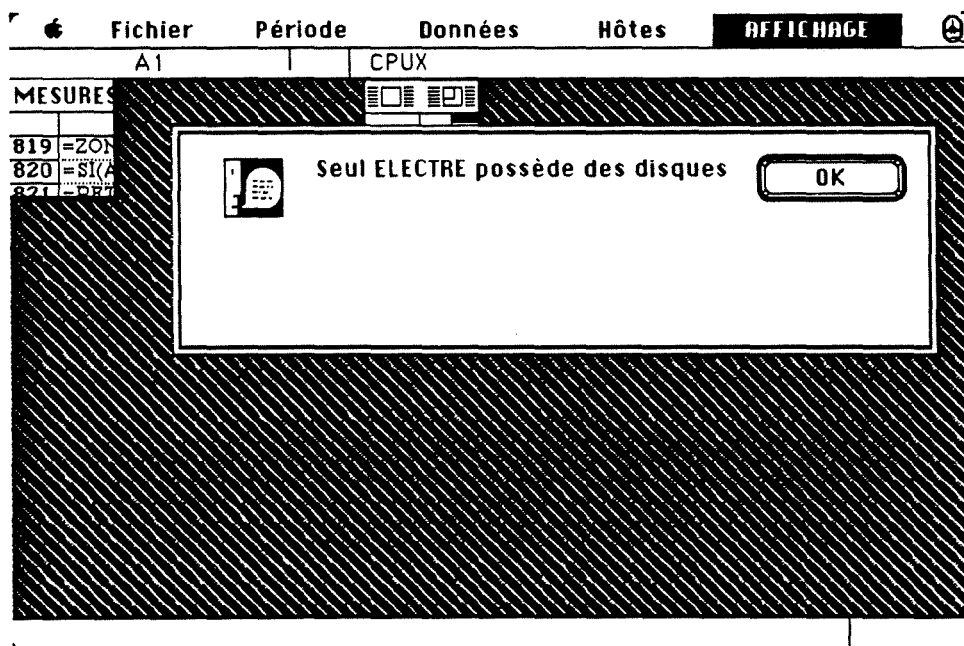
Il reste à l'utilisateur à déterminer l'hôte sur lequel les mesures affichées porteront. Les hôtes sont groupés par serveur dont ils dépendent. Un premier choix doit être fait entre ces différents groupes à partir du quatrième menu.



Une fois ce choix effectué, une zone de dialogue propose un choix entre les différents hôtes faisant partie de ce groupe.

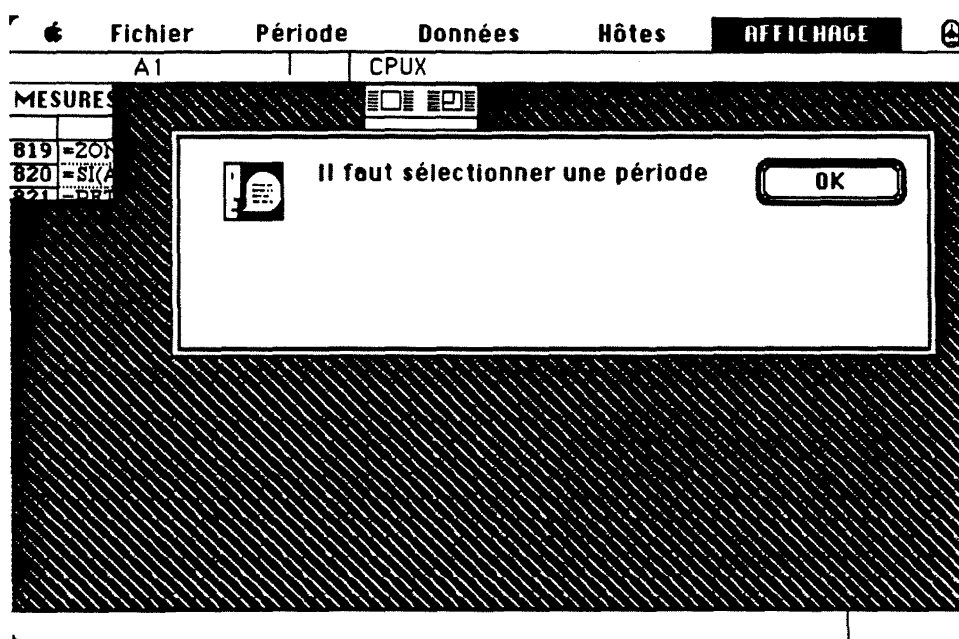


Si l'utilisateur a choisi des mesures concernant l'utilisation des disques et un hôte qui n'est pas équipé de disques, un message d'erreur l'avertira de l'incompatibilité de ses choix.

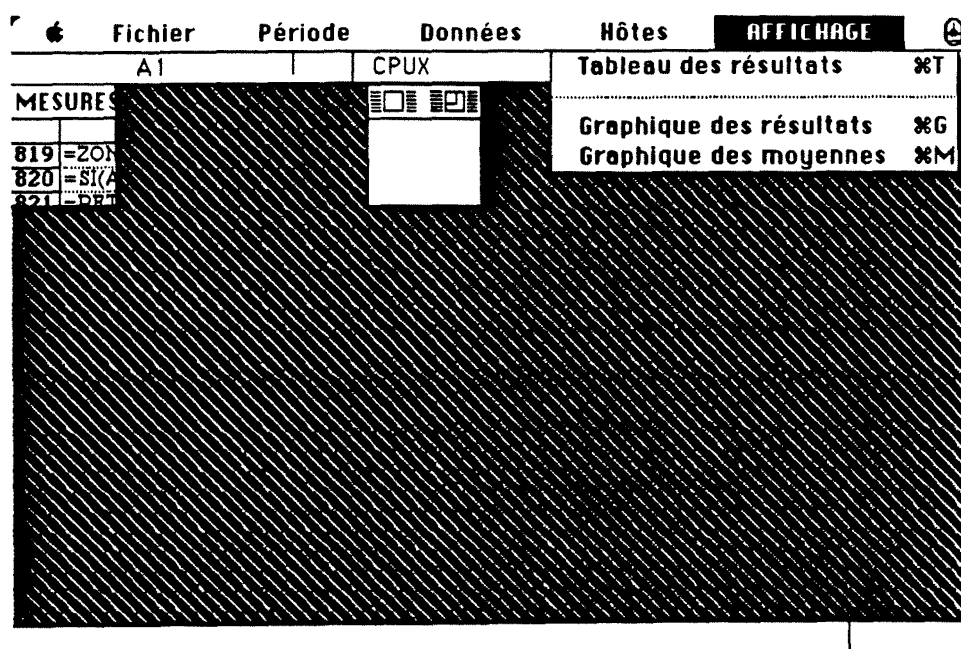


3.4.2.3. CONSTRUCTION DES TABLEAUX ET GRAPHIQUES

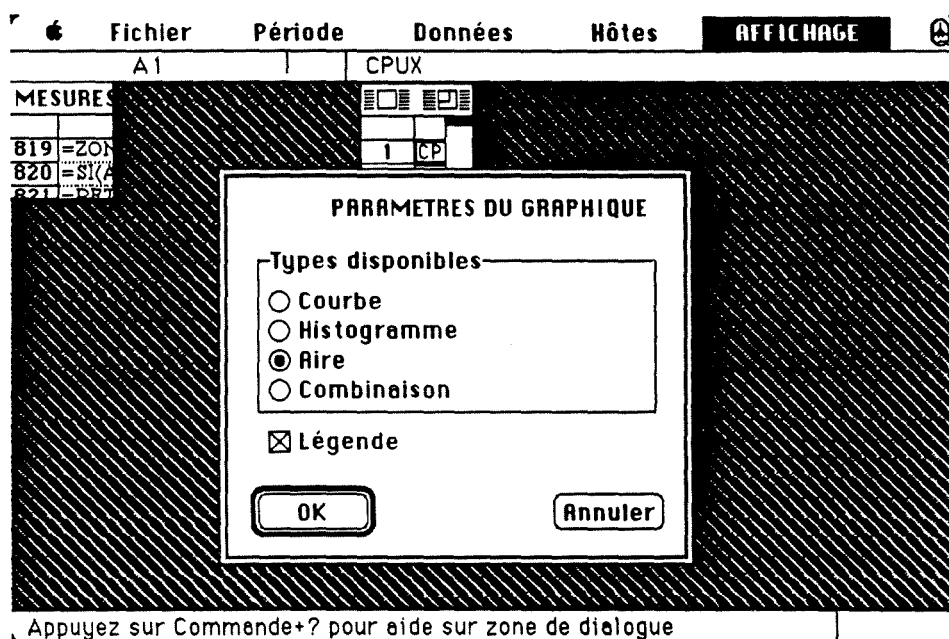
Une fois tous les choix effectués, on peut procéder à l'affichage des mesures. Si une des sélections a été omise, un message d'erreur avertira l'utilisateur de son oubli.



Les mesures peuvent être affichées sous forme de tableau ou de graphique, il est également possible d'obtenir un graphique des moyennes. Ces choix se font à partir du menu affichage, une fois le choix dans ce menu effectué, l'affichage est lancé.



Lorsque l'affichage est lancé, une zone de dialogue permet la sélection du type de graphique souhaité et la présence éventuelle d'une légende. La liberté est donc laissée à l'utilisateur de choisir la présentation la plus adaptée au type de mesures sélectionné.

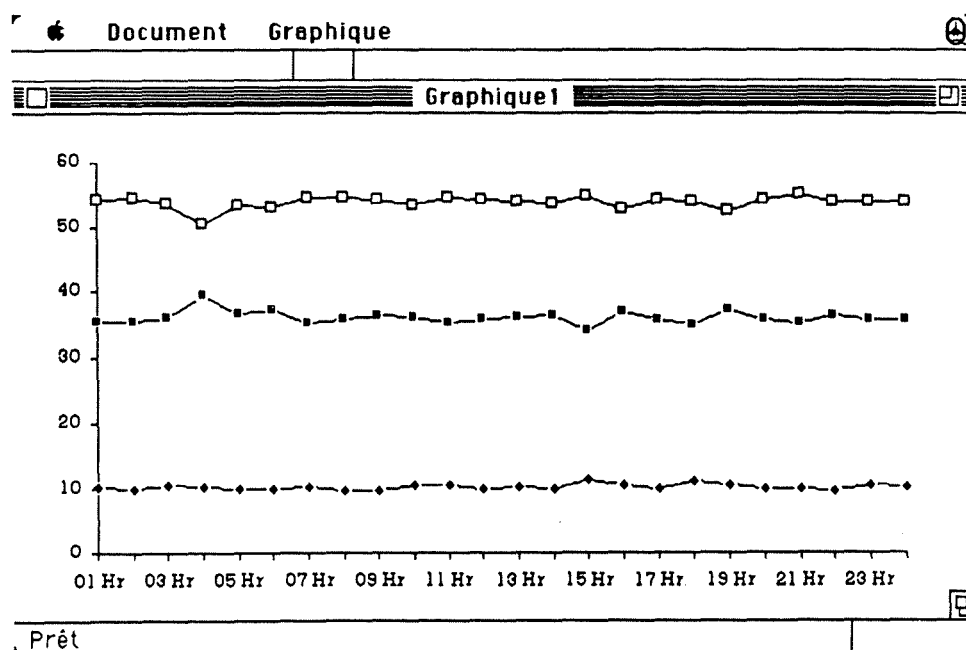


L'exécution de l'affichage graphique a d'abord pour effet de construire le tableau des données à partir desquelles le graphique sera réalisé. Ce tableau est identique à celui construit lors de l'activation de l'affichage en tableau.

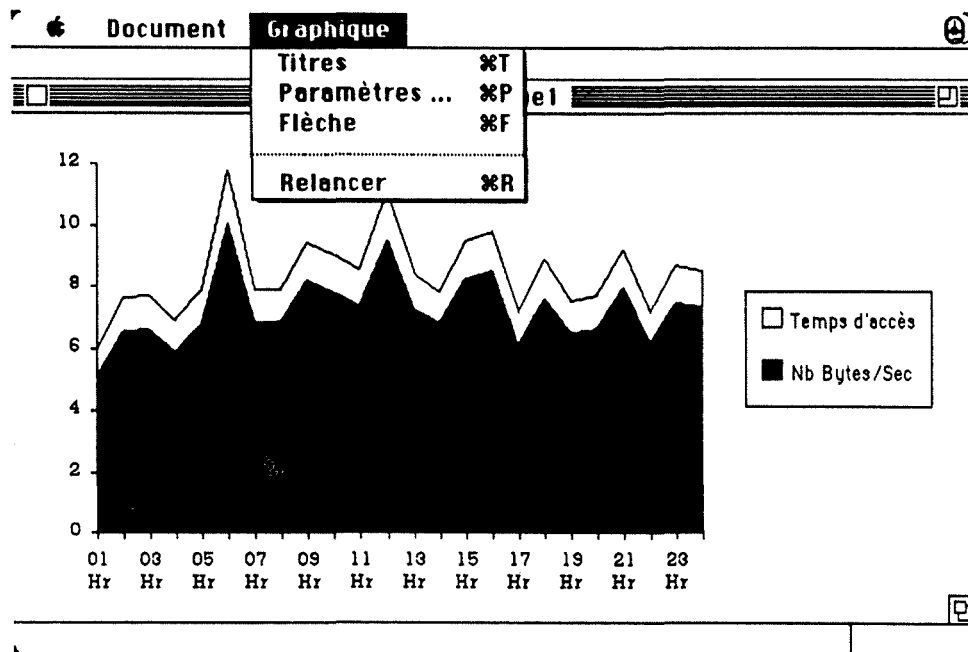
Fichier Période Données Hôtes AFFICHAGE						
A1						
Feuille de calcul1						
	A	B	C	D	E	F
1						
2	ELECTRE	Occupation du CPU				
3						
4		Utilisateur	Système	Endormi		
5	01 Hr	35,475	54,35	10,275		
6	02 Hr	35,55	54,6	9,95		
7	03 Hr	36,15	53,675	10,35		
8	04 Hr	39,4	50,625	10,075		
9	05 Hr	36,625	53,5	9,975		
10	06 Hr	37,125	53,1	9,95		
11	07 Hr	35,3	54,725	10,1		
12	08 Hr	35,75	54,75	9,625		
13	09 Hr	36,3	54,35	9,45		
14	10 Hr	36,1	53,575	10,475		
15	11 Hr	35,225	54,5	10,375		
16	12 Hr	35,775	54,45	9,875		
17	13 Hr	36,15	53,925	10,1		

Prêt

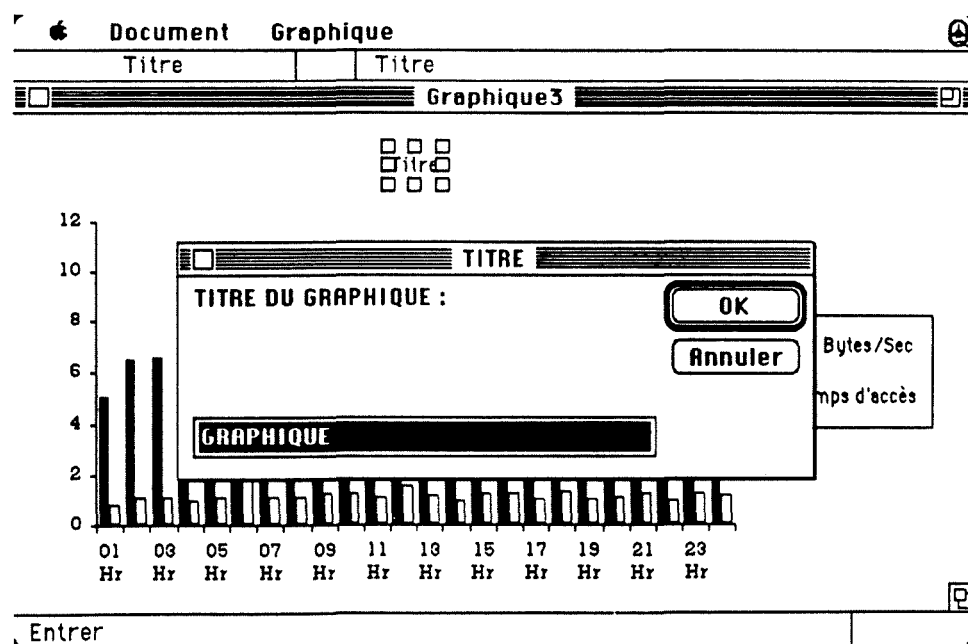
Le graphique se dessine d'une part sur base des paramètres fixés par le concepteur (format des échelles, polices de caractères,...), et d'autre part selon les choix effectués par l'utilisateur (type de graphique et légende).



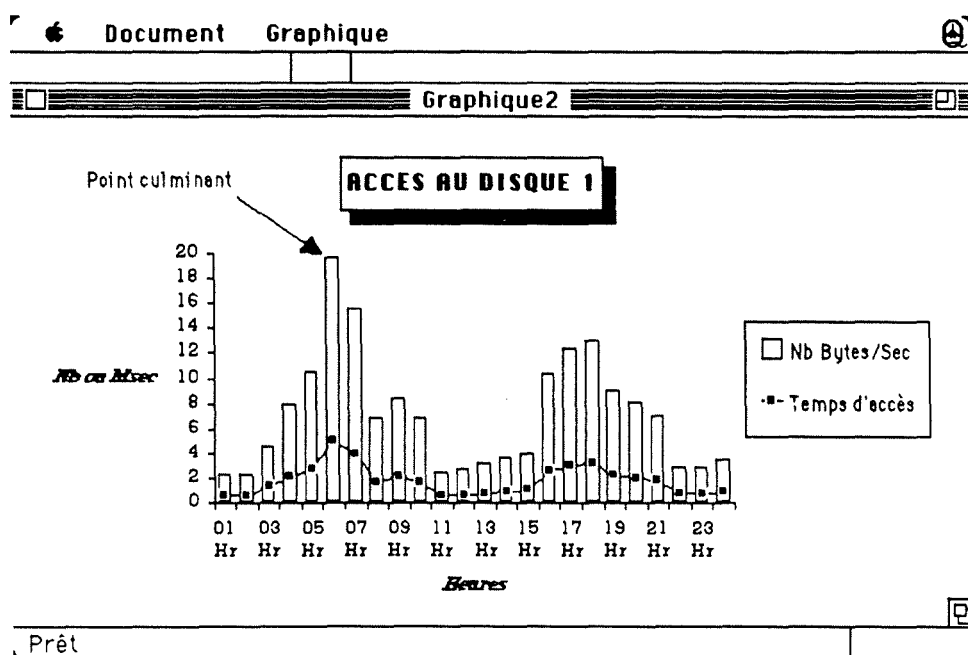
Remarquons qu'une nouvelle barre de menu est affichée en haut de l'écran. Deux menus sont accessibles : le menu fichier sur lequel nous reviendrons ultérieurement et le menu graphique.



Celui-ci offre quatre fonctionnalités à l'utilisateur. La commande Titre permet d'ajouter un titre au graphique ainsi qu'au deux axes, une zone de dialogue rend possible leur introduction.



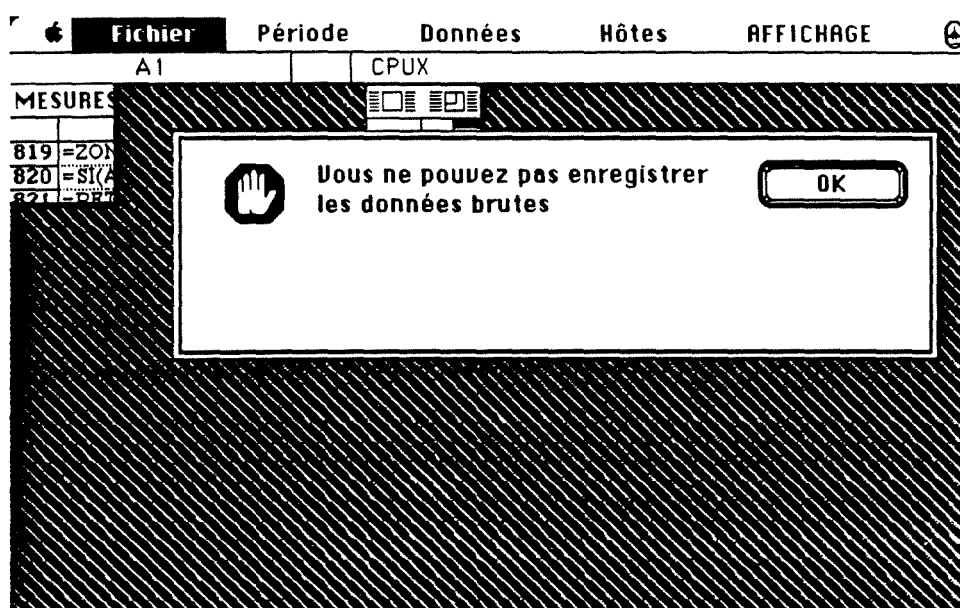
La commande paramètre permet une modification du type de graphique avant de redessiner le graphique sur les mêmes données au moyen de la commande Relancer. La commande Flèche offre l'opportunité de mettre en évidence un point particulier du graphique en ajoutant une flèche orientable accompagnée d'un commentaire.



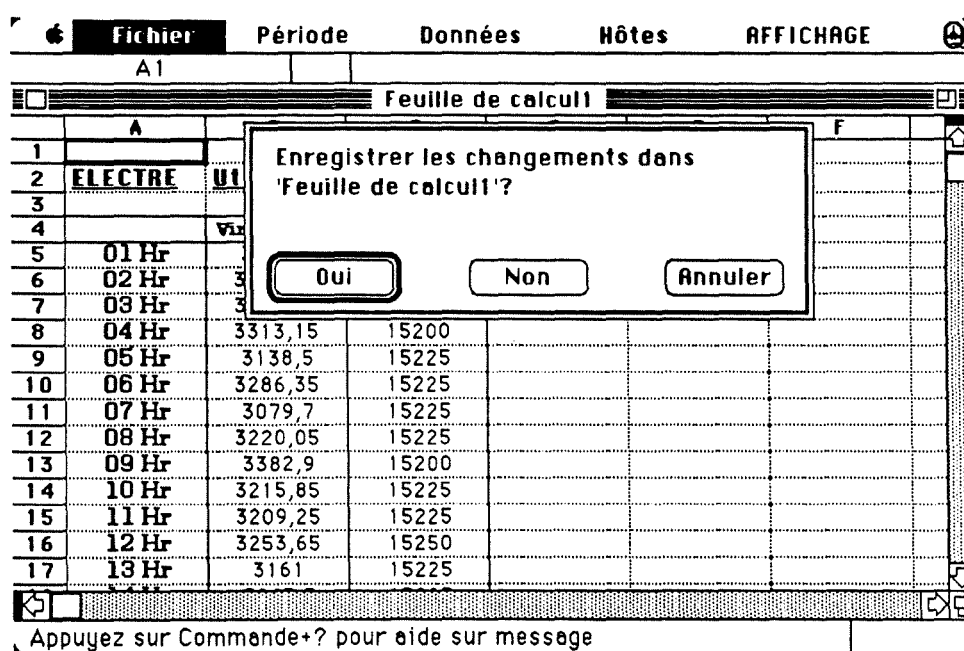
La commande fichier accessible tant en mode tableau qu'en mode graphique, offre des fonctionnalités proches de celles communément rencontrées dans les applications sur Macintosh.

Fichier				Période	Données	Hôtes	AFFICHAGE
Enregistrer ...				%E	Fille de calcul2		
Imprimer ...				%I			
Quitter Performance				%P			
Quitter ENCEL				%Q			
1					D	E	F
2					2		
3							
4							
5	01 Hr	Mb Bytes/Sec	Temps d'accès				
6	02 Hr	2,4	0,65				
7	03 Hr	2,4	0,65				
8	04 Hr	4,55	1,35				
9	05 Hr	8,025	2,225				
10	06 Hr	10,475	2,725				
11	07 Hr	19,6	5				
12	08 Hr	15,5	4				
13	09 Hr	6,825	1,75				
14	10 Hr	8,425	2,15				
15	11 Hr	6,9	1,775				
16	12 Hr	2,5	0,65				
17	13 Hr	2,725	0,725				
18		3,2	0,85				

Elles sont toutefois réduites à celles qui assurent la réutilisabilité des résultats. Le menu Fichier en mode tableau offre quatre fonctionnalités dont deux sont accessibles uniquement à ce stade de l'application. Enregistrer et Imprimer ne sont activables que si un tableau est actif à l'écran. Une tentative d'activation à un autre moment occasionne un message d'erreur.



La fonction Quitter Performance permet de sortir du logiciel MESURES, mais de rester dans l'environnement Excel. Cette fonctionnalité est utile pour récupérer le tableau en le sélectionnant et en le copiant dans le presse-papier, en vue de sa récupération dans une autre application. La fonction Quitter Excel a le même effet que la commande Quitter de toute application sur Macintosh, en ce compris les demandes de sauvegarde écran.



3.5. UTILISATION DE L'OUTIL

3.5.1. PERTINENCE ET VALEUR DES MESURES

Les mesures collectées par l'outil dont nous avons expliqué le développement et l'utilisation en détails doivent, pour constituer une aide efficace à l'administrateur du réseau, être non seulement correctes mais également pertinentes. La pertinence étant considérée dans ce contexte comme la faculté de permettre une interprétation des valeurs et d'avancer un embryon de diagnostic assez proche de la réalité. La correction des mesures représente quant à elle, leur valeur théorique. Il est nécessaire qu'elles soient théoriquement représentatives de la grandeur réelle qu'elles veulent mesurer. Nous allons aborder ces deux caractéristiques distinctement en évoquant toutes les mesures collectées par le logiciel.

3.5.1.2. CORRECTION DES MESURES

La correction des mesures est la première qualité qu'elles doivent remplir pour susciter un certain intérêt de l'administrateur, et ce avant même d'aborder leur pertinence.

Un première remarque importante nous paraît devoir être signalée avant d'aborder les différentes mesures en détails. Les résultats présentés à l'utilisateur constituent toujours des moyennes, soit journalières pour le fichier du mois courant ou du mois précédent, soit horaires pour le fichier du jour courant ou précédent. La moyenne quotidienne est extraite des 24 moyennes horaires, chacune d'elle étant calculée sur base de 4 prises de mesures séparées par un intervalle de 15 minutes. Le choix de l'intervalle de veille de 15 minutes entre les activations du mesureur repose, comme nous l'avons déjà évoqué, sur un équilibre entre deux critères essentiels, d'une part l'impossibilité de mesurer le système en permanence sans le "charger" artificiellement de manière excessive et d'autre part la nécessité de ne pas trop espacer les mesures afin de pouvoir tenir compte d'un maximum d'événements. Le choix de 15 minutes est en fait motivé par l'utilisation d'un délai de cette grandeur dans certains des logiciels existants et par la présence de ce délai dans le système lui-même et notamment dans la notion de charge moyenne. Nous allons donc nous attarder plus en détails sur la manière dont ces mesures ont été prises lors des activations toutes les 15 minutes.

Trois méthodes de collecte utilisées dans les modules de prise de mesure et de filtrage nous permettent de définir trois types principaux de mesures : les mesures cumulatives, les mesures ponctuelles moyennes et les mesures ponctuelles.

Les mesures ponctuelles peuvent être définies comme celles qui sont extraites une seule fois du système au moment de la prise de mesure et dont la forme peut être soit une mesure instantanée, soit une moyenne calculée par le système avant de la fournir. Ce type de mesure est certainement le moins intéressant et le moins correct au sens théorique du terme, du moins dans sa forme instantanée. En effet, la prise de mesure instantanée est influençable par les événements ponctuels survenant durant le court instant de la mesure et dont l'impact fausse le profil général que devrait mesurer l'outil. C'est la raison évidente pour laquelle nous ne présentons aucune mesure de ce type, les seules mesures ponctuelles utilisées pour le calcul des moyennes horaires ou journalières ont déjà la forme d'une moyenne fournie par le système, tel est le cas du temps moyen de pagination et du temps moyen de réclamation de pages. Ces deux moyennes ne posent aucun problème de correction puisqu'elles sont extraites par le système des données cumulées depuis le dernier relancement du système. La seconde moyenne fournie de manière instantanée par le système est la charge moyenne de ce dernier, c'est-à-dire la longueur moyenne de la file d'attente au CPU. Cette dernière s'accommode particulièrement bien avec notre outil puisque qu'elle présente trois moyennes de charge, une durant la minute précédant la mesure, une durant les cinq minutes et la troisième durant les quinze minutes précédant la mesure. Il est donc évident que cette dernière est celle qui nous intéresse le plus puisque sa définition, combinée au principe de mesure et au calcul des moyennes opérées par notre outil, permet de couvrir tous les événements survenant endéans l'heure de référence. Une troisième mesure fait exception et constitue réellement une approche instantanée, mais qui n'entrave en rien sa correction vu son évolution très lente, il s'agit en fait de l'occupation relative des disques par les différentes partitions.

Certaines données fournies par le système et qui revêtent une importance capitale en matière de performance sont fournies sous une

forme instantanée. Afin de maintenir un degré de correction raisonnable en ne relativisant les événements exceptionnels momentanés, il est nécessaire de suppléer au manque de moyenne en la calculant soi-même. Plusieurs mesures sont acquises selon ce principe de base que nous allons développer en détails. Les mesures concernées sont celles décrivant l'état du processus, son pourcentage d'occupation, les accès aux disques et l'utilisation de la mémoire. Le principe des commandes Unix utilisées pour extraire ces commandes nous permet de stipuler deux arguments importants : le nombre d'échantillons à prélever et l'intervalle les séparant. Nous avons décidé d'exécuter les commandes de mesures en utilisant cette possibilité et en définissant dix échantillons séparés d'un intervalle d'une seconde. Ce choix permet d'une part de relativiser les écarts dus à certains événements et d'autre part de ne pas surcharger artificiellement le système pendant trop longtemps. Le choix de dix secondes est un peu subjectif; on aurait pu déterminer un nombre plus grand tel que 15 ou 20 mais, suite à plusieurs essais, la probabilité qu'il soit à coup sûr plus correct nous semblait minime. L'intervalle d'une seconde nous paraît quant à lui excellent pour suivre le système pendant un certain temps, il représente en quelque sorte la notion commune de temps perceptible tant par l'ordinateur que par le monde extérieur. Ces dix mesures successives sont, pour chaque donnée, ramenée à une moyenne par le premier module de filtrage, ce dernier fournissant ainsi, au module de calcul de la moyenne horaire, une valeur plus correcte qu'une prise instantanée.

Une grande partie des mesures prises par le système peut être accédée sous forme de données cumulatives. Le point de départ de la sommation des valeurs étant le dernier relancement du système. Cette possibilité offerte par certaines commandes de Unix nous permet de présenter des mesures moyennes correctes mais dont la courbe d'évolution est, comme pour les deux autres types évoqués, "lissée" par période d'une heure. Les mesures concernant le trafic à travers l'interface du réseau, l'activité de pagination à l'exception de la mesure évoquée plus haut et les mesures de certaines activités du CPU (nombre d'interruptions, d'appels au système et de changements de contexte du processeur) sont saisies selon cette technique. Nous avons choisi de présenter le total par heure, sans les ramener à la minute ou à la seconde pour deux raisons principales. D'une part, la rigueur de la correction d'une présentation à

l'heure est sans équivoque, la mesure ne subit aucune généralisation. Ce qui serait par contre le cas si on ramenait le résultat à la seconde par exemple. D'autre part cette solution permet de laisser à l'administrateur la liberté de l'interprétation et de la généralisation.

3.5.1.2. PERTINENCE DES MESURES

Maintenant que la correction des mesures ne pose plus de problèmes, il faut prouver que ces mesures correctes sont utilisables et interprétables. L'intérêt des mesures réside dans cette qualité et, si elle est absente, la motivation de la mise au point d'un tel outil est totalement compromise. Pour amener le lecteur à se convaincre de l'utilité des mesures collectées, deux approches se complètent assez bien.

Tout d'abord le choix des données à sonder ne s'est pas basé sur un hasard total mais bien sur une évaluation des besoins des gérants de système dont la fonction est d'analyser et d'interpréter de telles mesures. Leur expérience et leurs connaissances sont la première garante de l'efficacité de ce choix.

La seconde approche doit permettre de prouver l'interprétabilité des données, leur cohérence et leurs interrelations. Ces caractéristiques doivent permettre à l'administrateur du système de remplir efficacement sa fonction de manager et plus particulièrement d'opérer sur le système dont il est responsable une politique de **configuration Management** efficace et cohérente.

3.5.2. REALISATION DE L'ANALYSE

Le rôle de cette application est de fournir une étude du comportement et de la charge du serveur vax Electre et de la station WS1, cette dernière pouvant être considérée comme représentative. Cette étude analysera le comportement de ces machines au cours d'une journée de travail habituelle.

3.5.2.1. HYPOTHESES

Plusieurs hypothèses de départ doivent être émises, afin de garantir une uniformité dans l'interprétation et la compréhension des résultats.

La première hypothèse nous permet de considérer que la journée prise en compte correspond à une journée type de travail à l'Institut. Celle-ci ne comprend donc aucune activité particulière ou exceptionnelle occasionnant une déviation significative de l'activité et par la même des mesures prises. Nous avons en fait analysé neuf journées types et constaté que l'évolution générale des différentes mesures restait similaire d'une journée à l'autre.

Une deuxième hypothèse concerne le choix de la station WS1 comme poste de travail de référence. Après avoir comparé les données concernant les différentes stations, nous avons constaté que l'évolution de leur comportement était très proche à l'exception de la station WS2, cette dernière n'étant pas accessible comme les trois autres.

La troisième et dernière hypothèse tient compte de l'influence du mesureur lui-même sur les données qu'il collecte. Le mesureur étant exécuté par un processus se déroulant sur l'hôte faisant l'objet des mesures, nous considérons qu'il a une influence sur les mesures prises et notamment sur le nombre de processus, les accès au réseau, l'utilisation de la mémoire, etc. Nous émettrons l'hypothèse que l'influence de l'outil sur ces mesures est généralement négligeable sauf lorsqu'elle est facilement remarquable comme dans le cas du nombre de processus exécutés.

3.5.2.2. ANALYSE DES MESURES COLLECTEES

Cette analyse prendra en compte cinq domaines de performance tels qu'ils sont définis dans l'outil de mesure, à savoir l'utilisation du CPU, les accès aux disques, l'occupation de la mémoire, l'activité de pagination et la contribution au trafic du réseau. Les différentes mesures analysées seront définies au fur et à mesure de leur apparition dans l'étude des différents domaines.

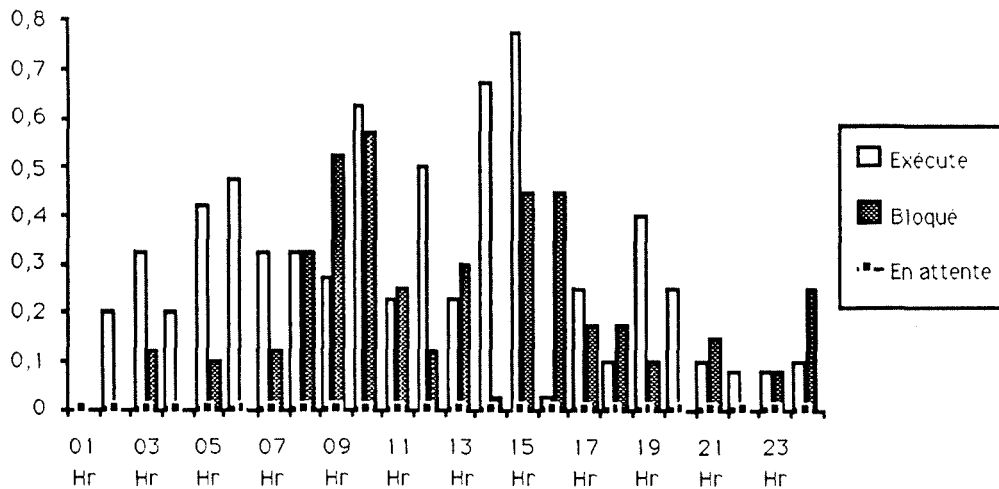
3.5.2.2.1. Utilisation du cpu

Les mesures concernant le CPU décrivent d'abord **le nombre de processus dans les trois états** suivants : dans la file d'exécution, bloqué par manque de ressource (entrée/sortie, pagination,...) et exécutable ou endormi pour une courte période (moins de 20 secondes) mais non résident en mémoire.

Les données concernant le serveur Electre et telles qu'elles sont représentées sur le graphique suivant montrent qu'aucun processus n'est en attente durant la journée et que le nombre de processus bloqués suit l'évolution des périodes de travail classiques. Il y a au cours de la journée en moyenne un processus bloqué, ce nombre atteint un maximum de 2 à 2,5 durant les périodes les plus chargées. Un processus en moyenne s'exécute sur le serveur, même durant les périodes "creuses", les processus système et notre outil de mesure sont vraisemblablement à l'origine de cette permanence. Les variations du nombre de processus en exécution sont très faibles (inférieures à 1).

La charge de travail semble donc garder une allure régulière durant les heures normales de prestation d'une journée.

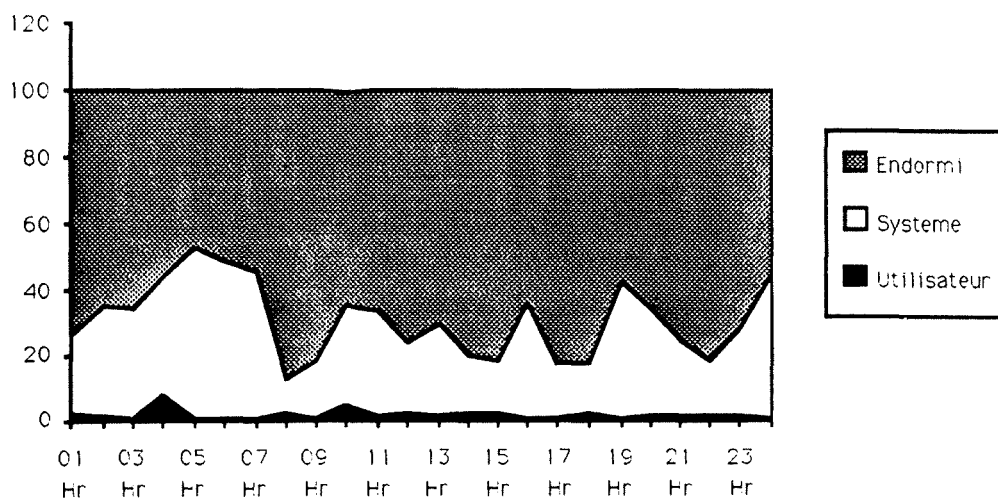
La charge de la station WS1 semble très limitée à première vue, en effet pratiquement aucun processus ne se trouve en état d'attente, alors que le nombre de processus exécutés ou bloqués oscille entre 0 et 0,8. Le graphique suivant illustre ces considérations.



nombre de processus de WS1

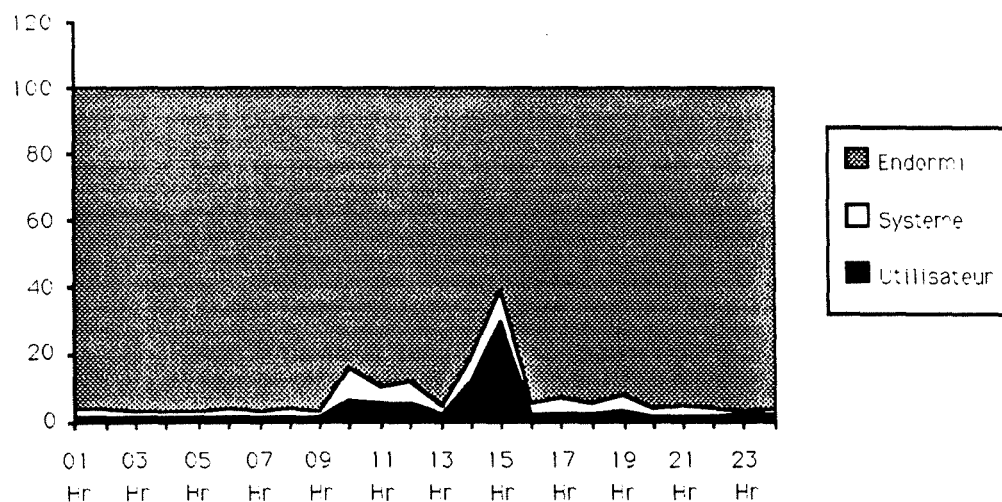
Rappelons néanmoins qu'il s'agit ici d'une mesure ponctuelle moyenne ne tenant, par définition, pas compte de tous les événements survenus dans l'heure de référence.

Le pourcentage d'utilisation du CPU dans les modes endormi, système et utilisateur constitue une mesure importante décrivant l'équilibre du système. Digital Equipment (voir annexe N) considère qu'un système bien équilibré doit passer la plupart de son temps en mode utilisateur (de 40 à 60 %).



occupation du processeur d'Electre

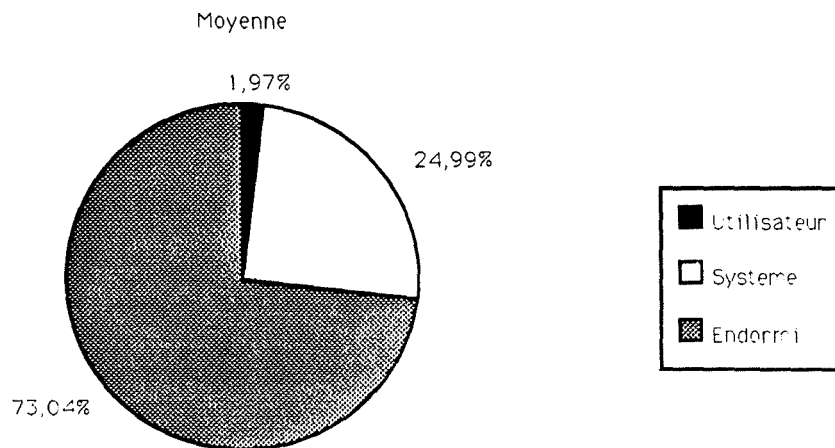
On remarque clairement que le taux d'utilisation du serveur Electre est faible, puisque le processeur passe de 60 à 80 % de son temps en mode endormi. Il consacre au mode système 20 à 40 % de son activité, alors que les utilisateur ne consomment que de 0 à 10 % de son temps. La finalité du serveur justifie l'importance du mode système par rapport au mode utilisateur, les stations de travail étant quant à elles consacrées principalement aux différents processus utilisateur. Les quelques pourcents passés en mode utilisateur par le processeur d'Electre proviennent vraisemblablement des processus activés par les utilisateur connectés à Electre soit via un terminal, soit via un autre hôte au moyen d'un remote-login. Vu le nombre de ports d'entrée limités à deux sur Electre, ce pourcentage devrait rester très minime.



occupation du processeur de WS1

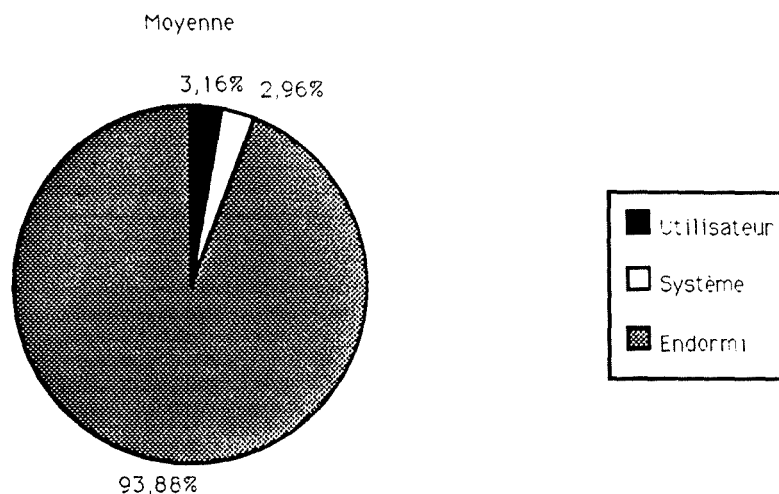
L'occupation du processeur de la station WS1 est représentative de son utilisation. Le processeur se trouve en mode utilisateur entre 0 et 30 % de son temps répartis entre 9 heures et 20 heures, ce qui correspond plus ou moins à la journée de travail. Cette correspondance est d'autant plus marquée que les deux zones de "forte" utilisation sont comprises entre 9 heures et 12 heures et entre 13 heures et 16 heures. Le fonctionnement en mode système oscille entre 0 et 10 % proportionnellement à l'évolution du mode utilisateur. On peut difficilement considérer qu'il s'agit d'un système

équilibré, puisqu'il passe entre 60 et 95 % de temps processeur en mode endormi (en fonction du moment de la journée). Ce déséquilibre se marque fort heureusement dans le sens positif mais témoigne néanmoins de la sous-utilisation des capacités du processeur de la station.



pourcentages moyens d'occupation du processeur d'Electre

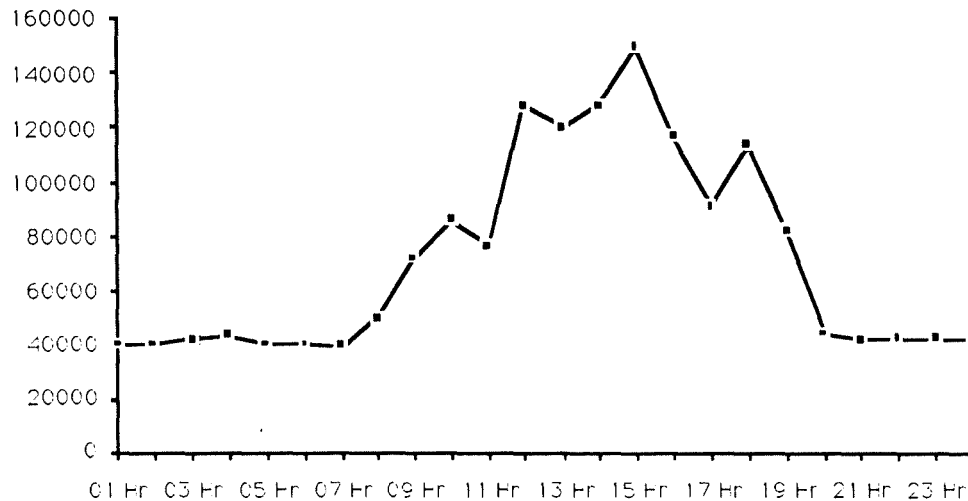
Les moyennes d'occupation du CPU, tant du serveur Electre que de la station WS1, sont significatives de leur type d'utilisation et constituent une approche relativement correcte de cette mesure.



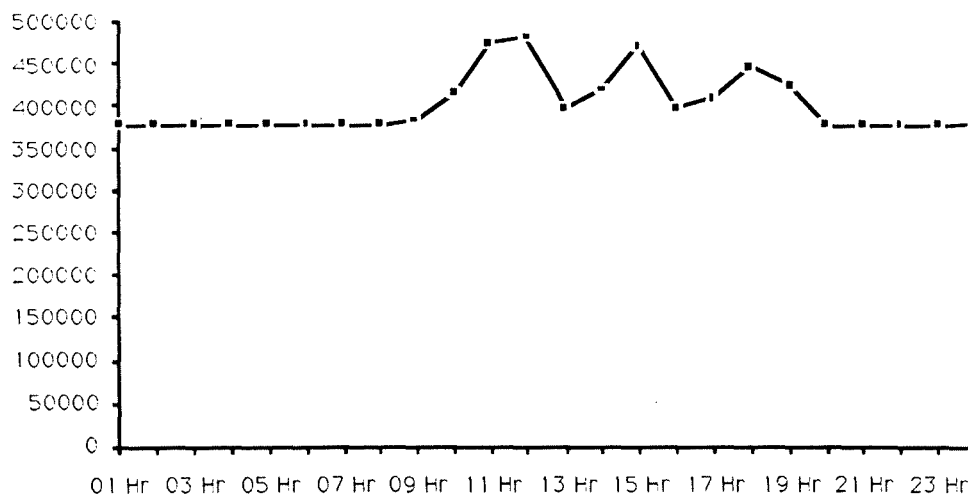
pourcentages moyens d'occupation du processeur de WS1

Les interruptions, les changements de contexte et les appels système sont également significatifs de l'activité du

processeur et varie proportionnellement à son taux d'occupation. Selon l'avis de DEC, un nombre élevé d'appels système est symptomatique d'une occupation trop élevée du processeur en mode système.



nombre de changements de contexte sur Electre



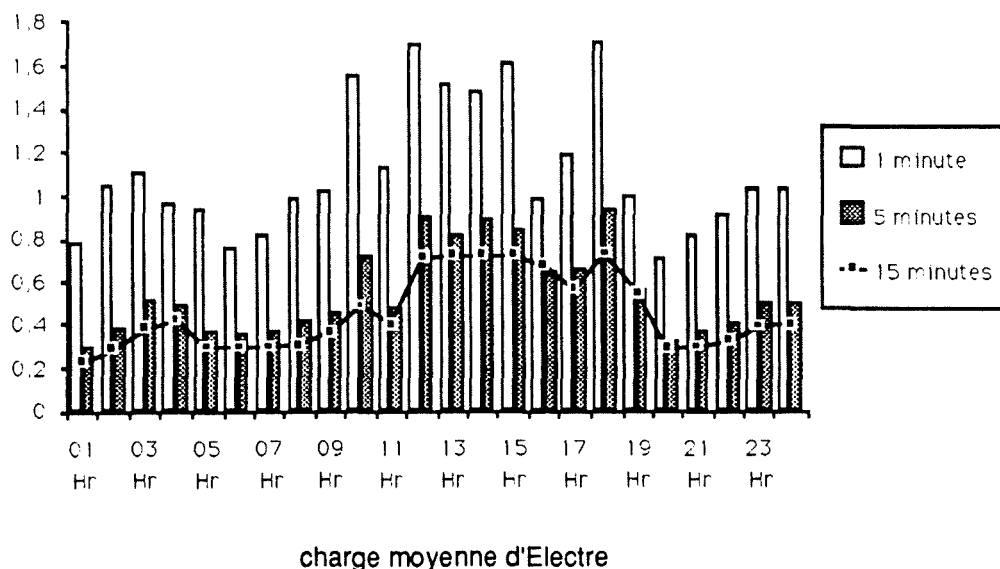
nombre d'interruptions sur WS1

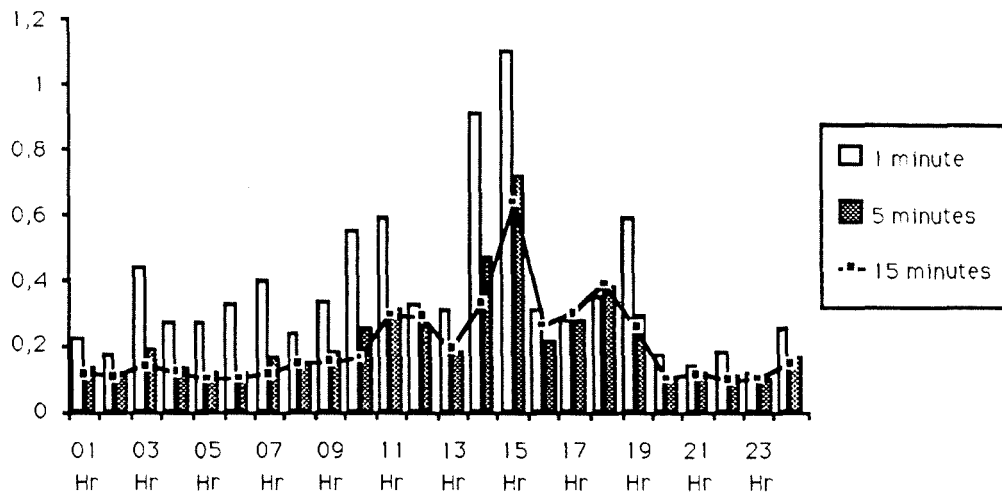
Les deux graphiques qui précèdent prouvent à souhait la relation entre l'activité du processeur et son taux d'utilisation.

La **charge moyenne** représente le nombre moyen de processus dans la file d'attente du processeur durant la dernière, les 5 dernières et les 15 dernières minutes. La charge moyenne n'est

pas extrêmement représentative de la charge globale d'une machine, elle dépend en effet de la proportion de processus orientés entrée/sortie ou CPU. Il est clair que dans un environnement où les processus sont quasi tous orientés calcul, la charge moyenne est beaucoup plus représentative que dans un environnement où les entrée/sortie dominant.

Les deux graphiques qui suivent illustrent parfaitement cette dépendance de la charge moyenne par rapport au type de processus exécutés sur la machine. Electre jouant le rôle de serveur exécute principalement des processus orientés entrée/sortie, ce qui a pour conséquence d'interférer beaucoup moins sur sa charge moyenne, celle-ci variant selon le nombre de processus système ou utilisateur en cours d'exécution. La station WS1, par contre, de par son rôle de processeur dévolu à l'utilisateur, exécute vraisemblablement un plus grand nombre de processus orientés calcul. Ceci pourrait expliquer la plus forte variation de la charge moyenne.





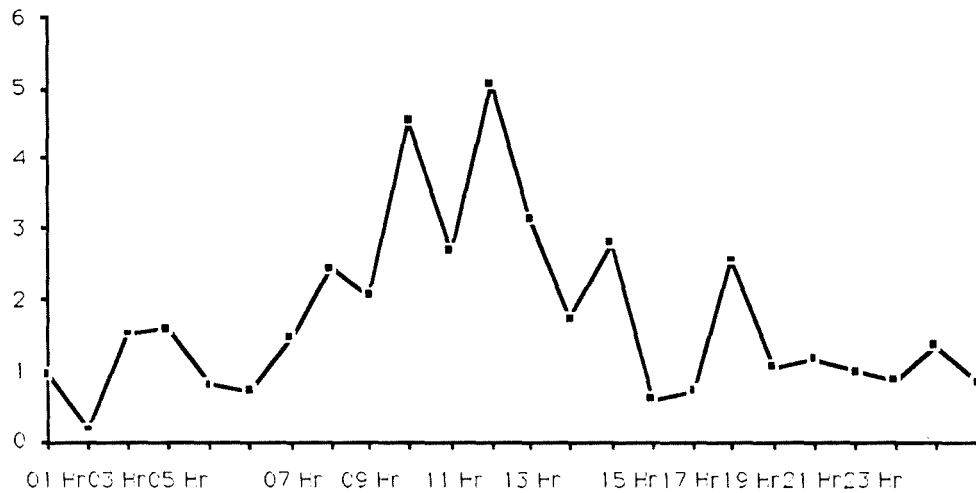
charge moyenne de WS1

L'intérêt de la charge moyenne durant les 15 dernières minutes est naturellement plus grand, car il est moins influencé par les événements ponctuels. Ceci explique le caractère plus "lissé" de la courbe représentant la charge moyenne des 15 dernières minutes. Celle-ci s'intègre parfaitement à l'intervalle de prise de mesure de l'outil. La moyenne des 4 charges moyennes permet donc d'obtenir une valeur moyenne horaire correcte puisque tous les événements sont pris en compte. Lorsque l'on analyse les grandeurs mesurées sur les deux hôtes, nous constatons que nous sommes loin de ce que DEC considère comme une forte charge moyenne (12 sur un vax 11/780, 6 sur un microvax et 20 sur un 11/750).

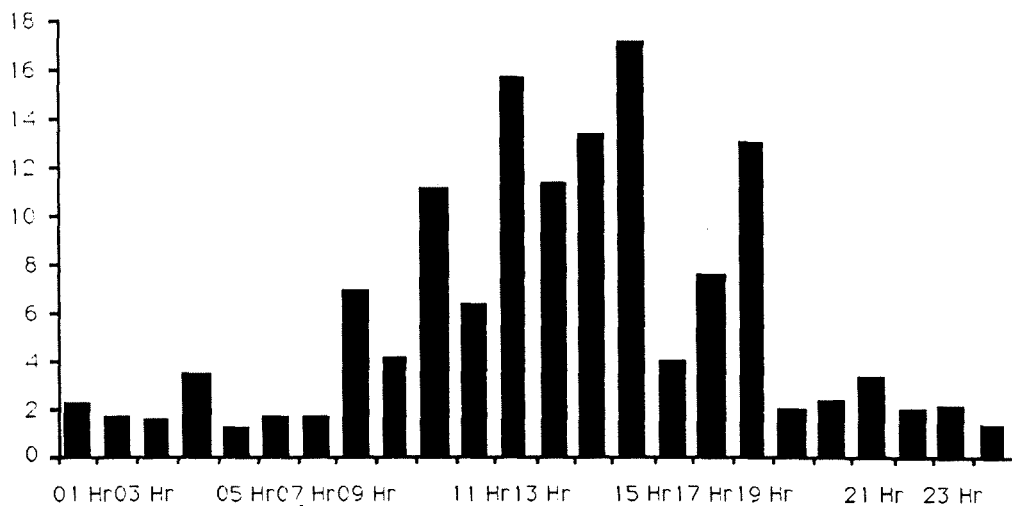
3.5.2.2.2. Accès au disque

Les mesures d'accès au disques peuvent s'exprimer selon trois grandeurs : **le nombre d'opérations par seconde, le nombre de bytes transfères par seconde et le temps moyen de recherche.** Ces grandeurs sont en relation étroite; une augmentation du nombre d'opérations par seconde a pour effet d'augmenter le nombre de bytes transfères par seconde et par conséquent le temps de recherche. Nous nous attarderons plutôt sur une comparaison des deux disques du serveur Electre.

Les graphiques suivants illustrent cette comparaison sur base du nombre d'opérations d'accès à ces disques.



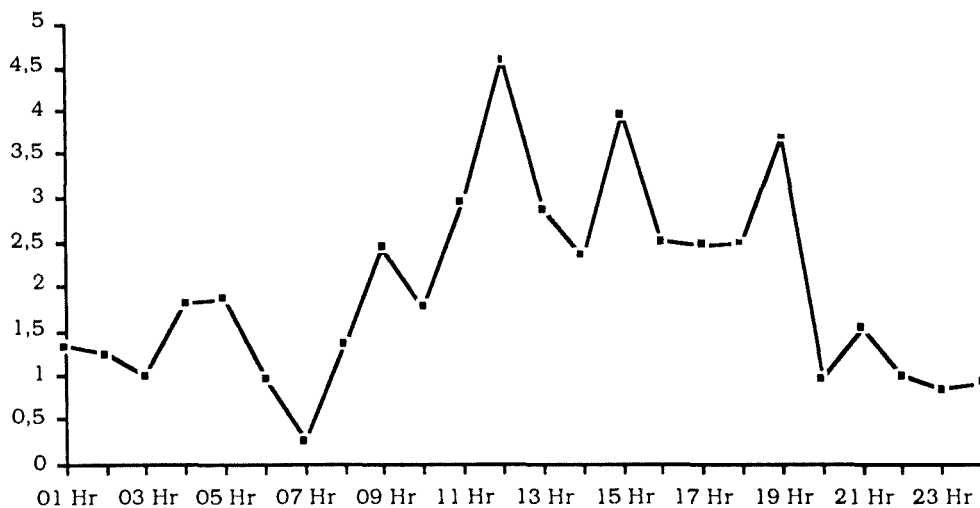
nombre d'opérations d'accès au disque 1



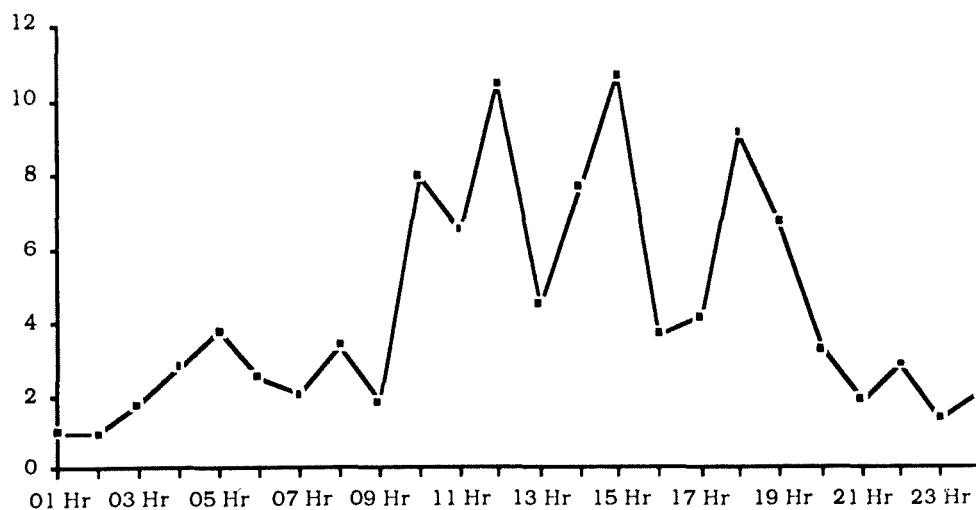
nombre d'opérations d'accès au disque 2

Le nombre d'opérations d'accès à l'un ou l'autre disque est bien sûr fonction du taux d'occupation des stations qui ne disposent d'aucun disque, c'est-à-dire en réalité du nombre d'utilisateurs. Cette dépendance se remarque assez clairement sur les deux graphiques avec un accroissement du nombre d'opérations durant les heures de travail.

L'accès au disque est aussi influencé par son contenu, la gestion de l'espace disque doit être judicieusement pensée afin d'éviter un goulot d'étranglement sur un disque alors qu'un autre est sous-utilisé. Le disque 1 contient l'espace de pagination du serveur lui-même ainsi que la partition /usr et /users/students, alors que le disque 2 regroupe l'espace de pagination des différentes stations. Il est donc logique de constater que le nombre d'opérations d'accès au disque 2 est supérieur au nombre d'accès sur le disque 1, l'activité de pagination étant plus importante sur les stations de travail que sur le serveur. On peut également constater, pour les mêmes raisons, un temps de recherche significativement plus élevé pour le disque 2.



temps moyen d'accès au disque 1



temps moyen d'accès au disque 2

La moyenne des temps d'accès au disque 1, calculée sur l'ensemble de la journée, tourne aux alentours de 2 millisecondes,

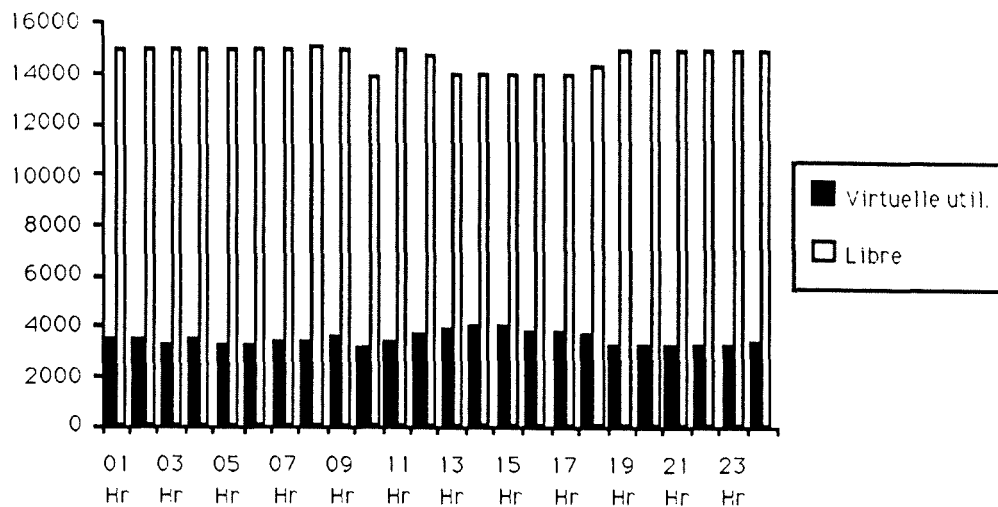
contre environ 7 millisecondes pour le disque 2. Ces valeurs, si elles paraissent peu élevées, s'expliquent par le fait qu'il s'agit d'accès au contrôleur de disque, celui-ci disposant d'une anté-mémoire. Le temps d'accès réel peut donc être bien inférieur au temps d'accès moyen complet (comprenant aussi le délai rotationnel) de 34,2 millisecondes renseigné par le constructeur.

3.5.2.2.3. Occupation de la mémoire

Il a souvent été dit qu'il était possible de résoudre tout problème de performance en ajoutant de la mémoire. D'après DEC c'est un peu vrai pour tout vax et c'est probablement une bonne idée dans le cas d'un 11/750 de 4MG.

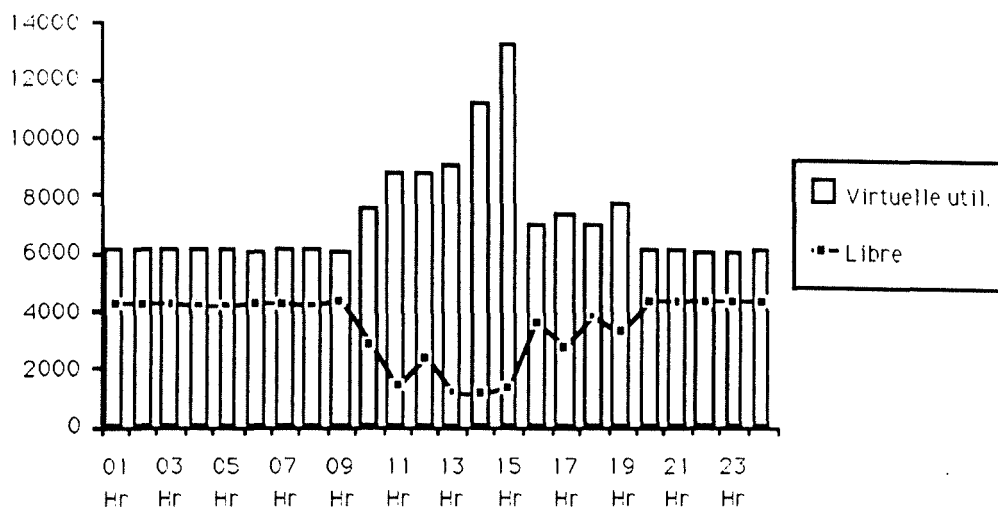
Au vu des deux graphiques qui suivent, la mémoire ne semble pas être le problème principal ni du serveur ni de la station. L'utilisation de la mémoire du serveur ne varie que très faiblement et la mémoire libre reste bien au delà de tout seuil critique puisque, selon DEC, il convient de s'inquiéter lorsque la mémoire libre tombe aux alentours de 500 pages.

La stabilité quasi-parfaite de la mémoire d'Electre colle parfaitement avec son rôle de serveur de fichier et la faible variation survenant durant les heures normales de travail pourrait peut être s'expliquer par les connexions directes d'un utilisateur à Electre soit depuis un terminal, soit depuis un autre hôte via un remote login.



Occupation de la mémoire d'Electre en kbytes

L'évolution de l'occupation de la mémoire de la station est naturellement beaucoup plus marquée que celle du serveur. L'augmentation de la mémoire virtuelle utilisée, logiquement parallèle à une diminution de la mémoire libre, se cantonne aux heures habituelles de travail. La diminution de la mémoire libre ne descend toutefois jamais sous le seuil des 1300 pages, ce qui écarte tout problème inhérent à un manque de mémoire.



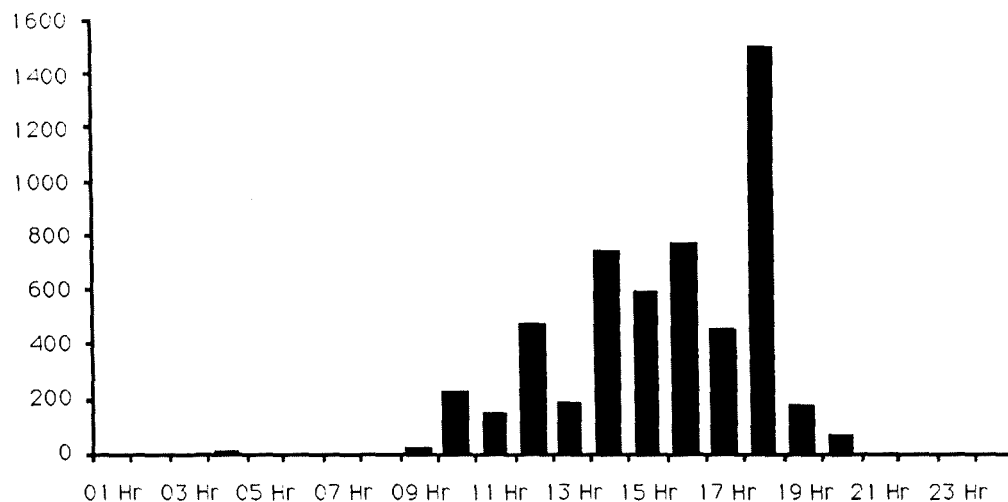
Occupation de la mémoire de WS1 en kbytes

3.5.2.2.4. Activité de pagination

La pagination est généralement une source de chute de performance, car elle est en relation étroite avec d'autres domaines comme la mémoire et les accès aux disques. Une chute des temps d'accès disque ou une insuffisance en mémoire, voir un cumul des deux circonstances, peut être à la base d'une activité de pagination excessive et par conséquent de chute des performances.

L'analyse de l'activité de pagination pour le serveur Electre n'offre que peu d'intérêt puisque d'une part il dispose d'une mémoire plus que suffisante et que, d'autre part, sa fonctionnalité première n'est pas de fonctionner comme processeur de travail, sauf dans les cas que nous avons déjà évoqués. Nous porterons donc notre attention sur l'activité de pagination de la station de travail, cet intérêt est d'autant plus justifié que la station que nous analysons est dépourvue de disque, la pagination se fait donc au travers du réseau augmentant encore ainsi son impact sur les performances.

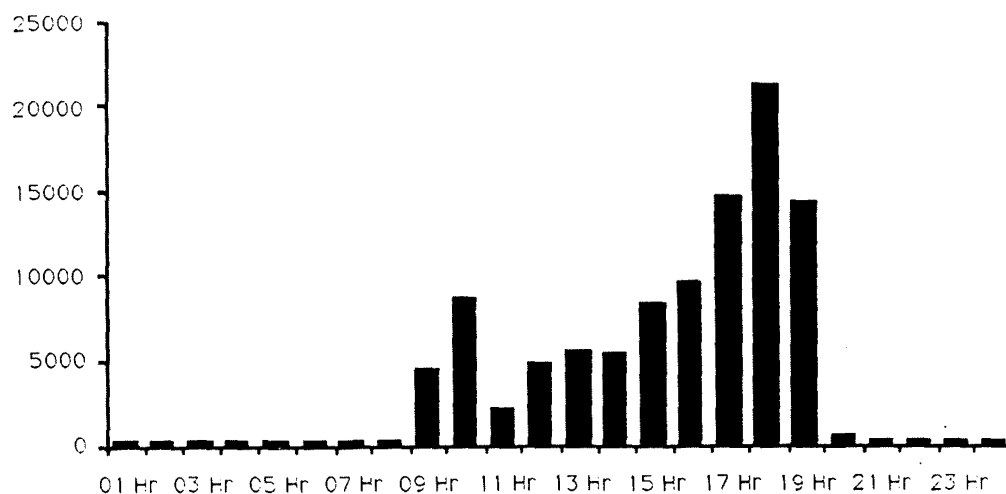
Le graphique qui suit met en évidence les défauts de page sommés pour chaque heure de la journée. On constate que l'activité est ciblée par les périodes de travail, c'est-à-dire entre 9 heures et 20 heures. A l'exception du point culminant de 18 heures, le maximum avoisine les 600 défauts de pages à l'heure, ce qui représente 10 défauts de pages à la minute, chiffre qui ne semble pas excessif à première vue.



Défauts de page sur WS1

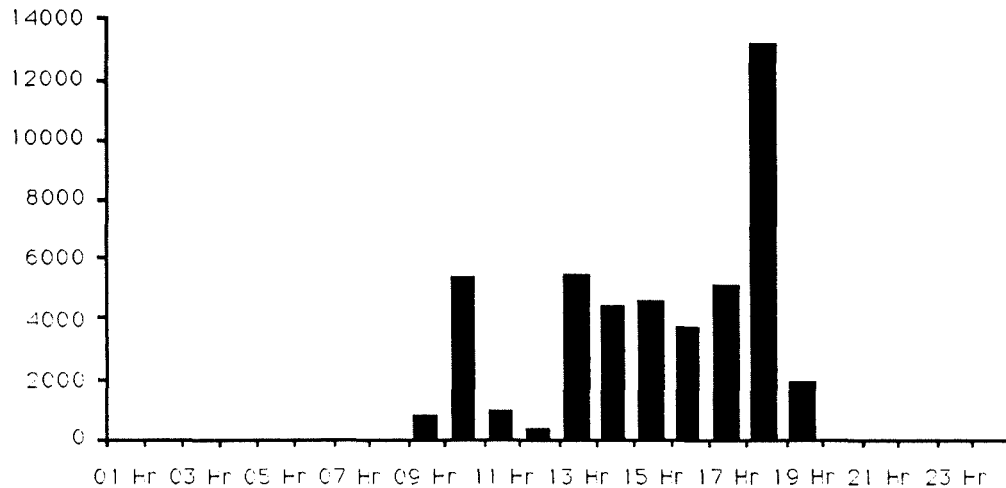
Les pages en entrée et en sortie mesurées correspondent aux pages échangées entre la mémoire et le disque pour assurer le déroulement d'un processus et non aux pages déplacées en raison d'un blocage du processus.

Les pages en entrée sur la station suivent également l'évolution de la charge de travail et sont relativement bien réparties sur la journée. On observe une moyenne de 4300 pages par heure sur une journée de travail, pour une moyenne de plus ou moins 12000 pages par heure pendant les heures de travail, soit 200 pages à la minute. En dehors des heures d'affluence le nombre de pages fluctue aux environs de 280 pages à l'heure.



Pages en entrée sur WS1

La même constatation peut être faite au sujet des pages en sortie. La moyenne horaire d'une journée tourne autour de 1900 pages alors que la moyenne horaire pendant les heures de travail peut avoisiner les 4800 pages, ce qui donne plus ou moins 80 pages à la seconde.



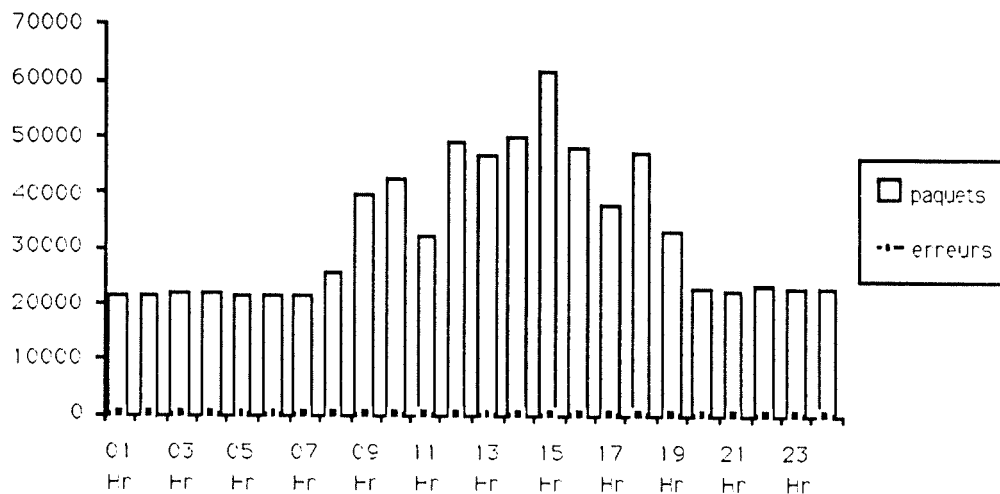
Pages en sortie sur WS1

3.5.2.2.5. Contribution au trafic du réseau

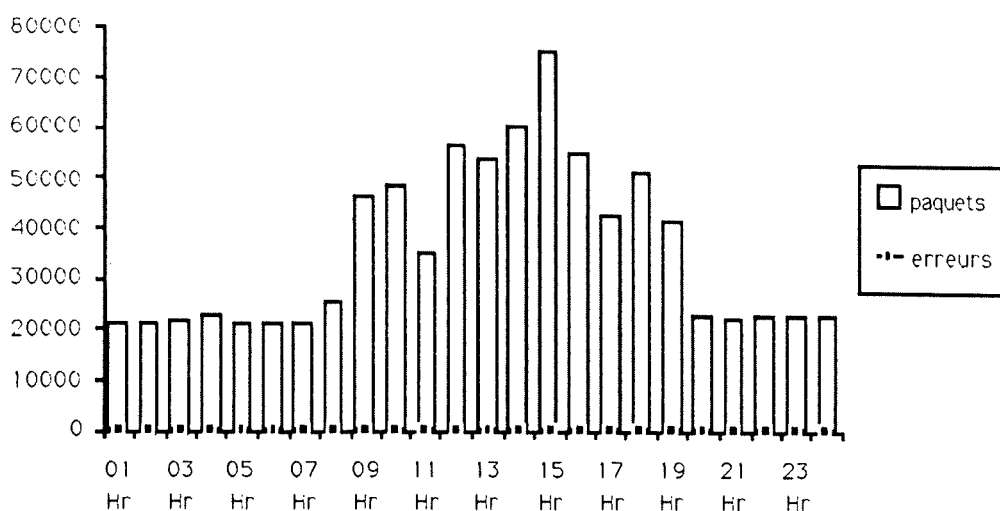
La contribution au trafic du réseau est une mesure importante, car elle permet de déterminer, en sommant soit les entrées soit les sorties de tous les hôtes connectés, la charge du réseau en paquets par unité de temps.

Ici encore nous allons constater, par chance, que l'activité de l'interface au réseau suit l'activité journalière des utilisateurs. En dehors des heures normales comme en pleine nuit, l'interface ne reste pas inactive. On peut expliquer ce trafic d'un part par des processus assurant la gestion du système (gérant de la mémoire, ...) et d'autre part, dans une moindre mesure, par l'activité engendrée par le mesureur.

Les valeurs en entrée et en sortie sont sensiblement de la même grandeur, les moyennes horaire par jour fluctuant autour de 32000 à 35000 paquets. Il est remarquable de constater que le nombre d'erreurs détectées est nul, à l'exception d'une seule détectée entre 15 et 16 heures en entrée du serveur.



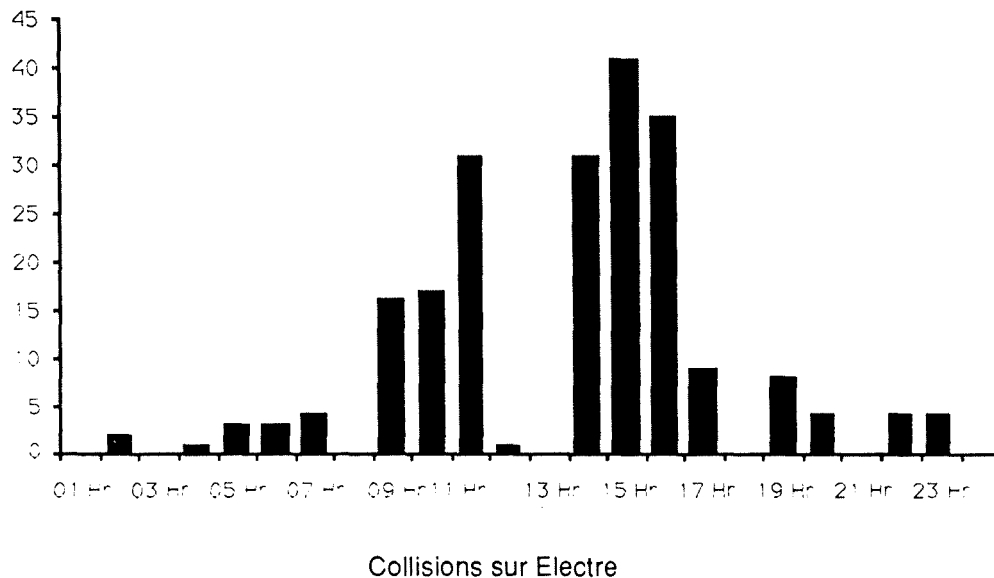
Paquets en entrée sur Electre



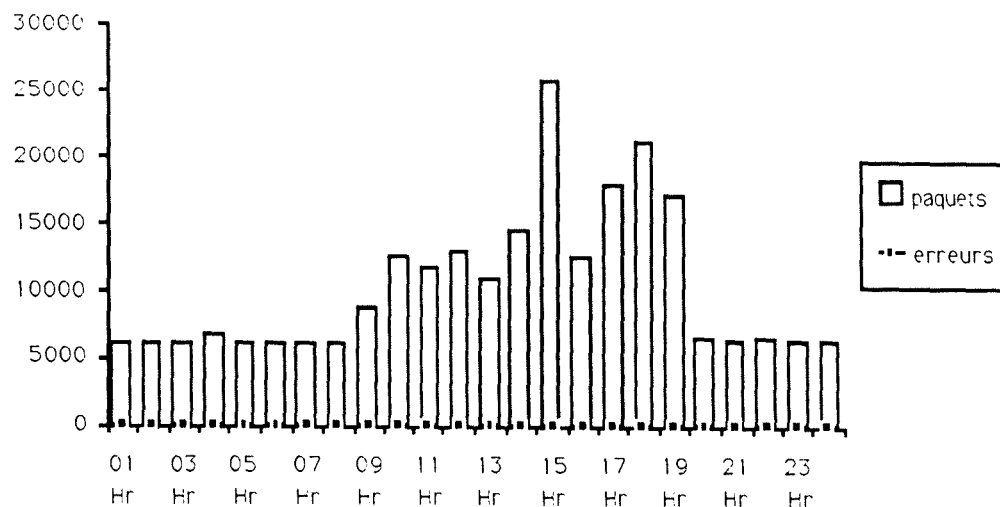
Paquets en sortie sur Electre

Les collisions sur le serveur Electre suivent une évolution presque parallèle au nombre de paquets traversant l'interface vers le

réseau. Il est curieux d'observer, à certaines heures, un phénomène de chute du nombre de collisions qui paraît a priori difficilement explicable.

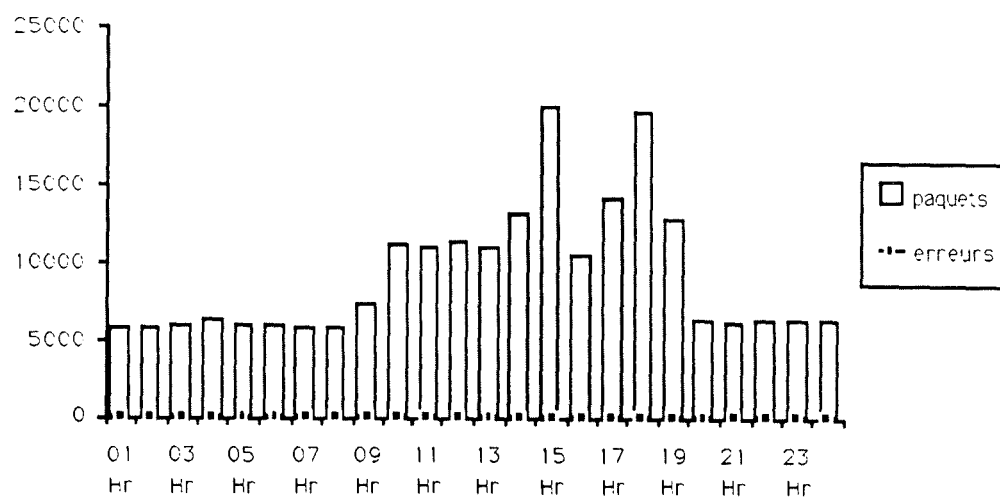


Le trafic traversant l'interface d'une station évolue d'une manière parallèle à celui décelé sur le serveur Electre. Cette évolution, si elle est semblable à celle constatée sur le serveur, se base sur des grandeurs trois fois inférieures, ainsi la moyenne horaire d'une journée tourne autour de 10000 paquets en entrée pour 32000 pour le serveur. Cette constatation n'est que logique si l'on réfléchit au fonctionnement du système. Le serveur supporte en effet au travers de son interface tous les paquets issus de l'activité de pagination de toutes les stations, les activités de sauvegarde et de lecture sur disque de ces même stations et les transferts de fichiers provenant ou allant vers d'autres systèmes de fichiers. Toute une partie de ce trafic, celui provenant des stations, se répartit plus ou moins équitablement entre ces dernières. Tout comme pour le serveur, les erreurs sont inexistantes tant en entrée qu'en sortie.



Paquets en entrée sur WS1

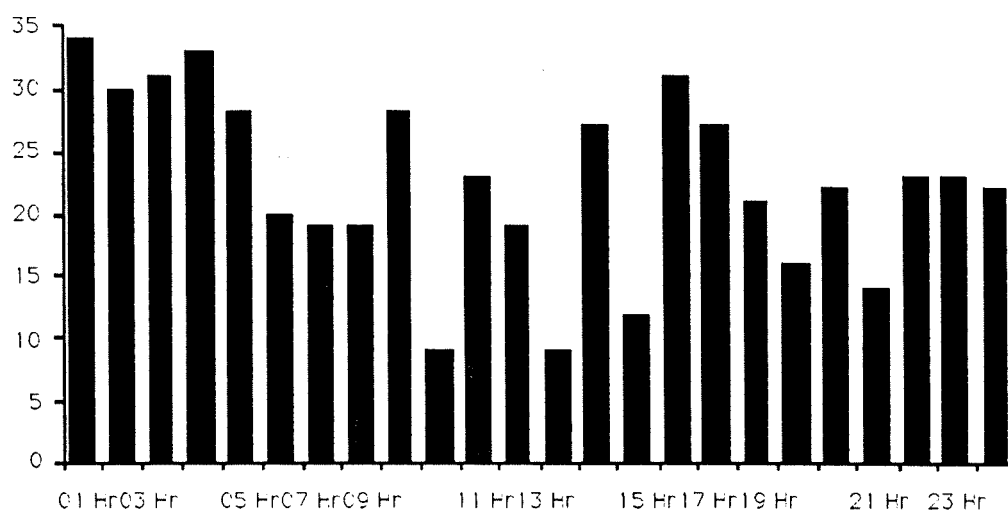
Les remarques évoquées au sujet des paquets en entrée sont d'application pour les paquets en sortie.



Paquets en sortie sur WS1

L'évolution des collisions sur la station se répartit selon une distribution qui paraît être totalement aléatoire. Le nombre de collisions semble assez important, durant la période de travail, proportionnellement à celle dénombrées sur le serveur pour un trafic trois fois supérieur. Nous avons dénombré sur Electre 15 à 40 collisions durant les heures de travail pour un trafic moyen situé pour cette période entre 80000 et 100000 paquets à

l'heure. Sur la station on peut constater la détection de plus ou moins 20 à 30 collisions pour un trafic tournant aux environs de 25000 à 30000 paquets à l'heure. Proportionnellement et en arrondissant, on peut affirmer qu'il y a trois fois plus d'erreurs par paquet que pour le serveur. Cette constatation est d'autant plus marquée aux heures creuses de la nuit, durant lesquelles on constatait la présence de 0 à 3 collisions pour plus ou moins 40000 paquets traversant l'interface du serveur, alors que le nombre de collisions signalées reste aussi élevé et parfois plus élevé que durant les heures de travail c'est-à-dire une moyenne de 30 collisions par heure pour un trafic de 12000 paquets, ce qui donne des chiffres de l'ordre de 30 fois supérieurs à ceux du serveur. Ces chiffres sont inexplicables.



Collisions sur WS1

3.5.2.3. ENSEIGNEMENTS

Cette analyse avait pour but de faire la lumière sur le comportement et la charge de la configuration Dec récemment acquise par l'Institut d'Informatique et composée d'un serveur Vax 3800 et de quatre Vaxstations 3200 tournant sous le système d'exploitation Ultrix 3.2 .

Le but que nous nous étions fixé était de fournir un rapport constituant un outil d'aide à la gestion du système, il devait fournir à l'administrateur une idée globale mais suffisamment précise du fonctionnement du système en temps réel. Nous avons analysé les différents domaines abordés par l'application de mesure et nous avons tenter d'en tirer les enseignements les plus utiles et d'établir pour chacun d'eux un profil de charge permettant d'évaluer la nécessité de modifier la configuration ou son organisation.

Nous pouvons nous rendre compte immédiatement que cette configuration ne présente aucun symptôme de surcharge. Le système semble pouvoir supporter un nombre plus important d'utilisateurs que les deux seuls que permettent tant le serveur que les stations. Les processeurs des stations sont loin d'être utilisés au maximum de leurs possibilités, ils passent en effet la plupart de leur temps en état d'inactivité. Cette constatation est aussi valable pour le processeur d'Electre qui ne semble nullement débordé par sa fonction de service auprès des différentes stations. Une augmentation du nombre de stations de travail ne compromettrait nullement la bonne tenue des performances du système. Il n'en serait surement pas de même s'il était utilisé d'une manière directe et intensive par des utilisateurs connectés, une baisse des performances des stations pouvant alors en résulter. La taille importante de la mémoire d'Electre contribue au bon comportement du système. Une amélioration du comportement des stations pourrait de même être obtenu en augmentant la taille de leur mémoire, réduisant ainsi leur accès au réseau et aux disques d'Electre.

Quoi qu'il en soit, les différentes ressources tant du serveur Electre que de ses stations répondent de manière plus suffisante aux demandes qui leur sont faites. Aucun indice de surcharge n'a été constaté dans l'ensemble des domaines qui ont fait l'objet de notre observation. Ce

rapport mettre donc en évidence la suffisance technique de la configuration telle qu'elle a été décrite et ses possibilités d'extension future.

Le choix par les administrateurs des mesures à prendre en compte, et l'application du logiciel à l'étude de la charge et du comportement des stations et du serveur qui vient d'être présentée, prouvent à souhait la valeur et la correction des mesures sélectionnées et collectées. Le logiciel mis au point revêt donc un intérêt manifeste pour le gestionnaire de la configuration, il constitue un outil efficace d'aide au **configuration management** .

3.6. CONCLUSIONS

Nous avons, tout au long de ce chapitre, suivi le cheminement de la conception d'un outil de mesure particulier à la configuration qu'il évalue. Nous avons basé cette démarche sur une étude de l'existant et une analyse des besoins tant globaux que spécifiques aux administrateurs des systèmes concernés. Notre analyse nous a amené à la conception complète d'un outil répondant aux souhaits émis par les différents gestionnaires et reprenant les avantages des outils existants, tout en évitant leurs inconvénients. L'utilité et la valeur des mesures collectées au moyen de cet outil ont été mises en exergue par son utilisation efficace dans le cadre d'une étude d'aide au configuration management réalisée sur l'évolution de la charge et du comportement des machines sur lesquelles l'outil est installé.

Ce logiciel étant réalisé et son utilité étant mise en évidence, il convient de tirer plusieurs leçons essentielles.

Les mesures de performances d'un système informatique doivent être en symbiose totale avec son cycle de vie. Une part essentielle de cette relation doit être consacrée au suivi d'un système en fonctionnement et à l'évaluation de son comportement et de sa configuration. L'allongement de la durée de vie d'une configuration et l'amélioration de son efficacité passe irrémédiablement par une connaissance générale de son comportement, de sa charge, de ses points forts et de ses faiblesses. L'intérêt d'un outil comme celui que nous avons développé rencontre donc parfaitement le besoin de connaissances d'un système et constitue une aide au maintien de cette symbiose importante.

Il existe plusieurs types d'outils permettant a priori d'assurer le suivi des performances d'un système, nous avons eu l'occasion d'en présenter deux disponibles sur le système supportant les mesures. Il convient de souligner à ce propos deux remarques importantes.

La finalité d'une majorité de ces outils ne répond pas, à notre avis, au besoin de connaissances nécessaires au suivi d'une configuration telle que nous l'avons définie au point précédent. Les deux outils présentés nous paraissent à cet effet représentatifs de ce problème, alors que paradoxalement ils sont très différents. Soit il s'agit d'un outil qui fait preuve d'une simplicité et d'une convivialité excellente mais qui se rapproche à notre avis plus d'un gadget que d'un logiciel d'aide au management du système. Ces outils n'offrent

de plus que rarement la possibilité de conserver leurs résultats car leur rôle est un suivi direct de la charge durant la session de travail. Soit il s'agit d'un outil permettant de sauvegarder des données importantes mais dont la complexité et la quasi absence de possibilité de réutilisation externe des résultats hypothéquent énormément l'efficacité pour un manager.

La différence de fonctionnalité et d'objectif dans les différents outils rencontrés doit être à notre avis l'expression parfaite de leur complémentarité, un outil de suivi à l'écran de la charge d'un système peut être considéré comme parfaitement complémentaire d'un outil tel que nous l'avons imaginé. Leur existence n'est donc pas concurrente mais convergente vers un objectif d'aide au management d'une configuration.

Les mesures de performance doivent être considérées non pas comme un but mais comme un moyen. Prendre des mesures sur un système n'a aucun sens si l'intérêt que l'on peut en retirer est nul, la raison d'être d'une mesure est de subir une analyse, une comparaison et une interprétation avec pour objectif à court ou à long terme de voir cette mesure évoluer dans le sens que l'on désire. Les mesures de performance doivent donc être à la base d'un processus d'analyse et de décision, approche rentrant parfaitement dans une conception managériale des mesures de performance telle qu'elle a présidé à la mise au point de notre outil.

Au terme de ce chapitre, nous espérons avoir suscité l'intérêt du lecteur pour les performances d'un système informatique et leur impact sur le processus décisionnel du manager qui le gère. Force nous est de constater que trop souvent encore cet aspect de l'informatique est mésestimé et tenu à l'écart de toute la gestion tant au moment de l'acquisition ou des améliorations que durant la vie du système. Puisse cette étude servir à faire naître ou croître cet intérêt pour les performances d'un système informatique et à faire prendre conscience aux administrateurs de système de la nécessité de disposer d'un outil leur offrant les mesures utiles à l'argumentation de leurs avis ou décisions.

CONCLUSIONS

Deux aspects primordiaux d'une configuration distribuée ont été envisagés dans ce mémoire : la sécurité et les performances. Nous avons abordé séparément ces deux problématiques au travers de deux approches méthodologiques tout aussi distinctes. Ce choix a eu le mérite de permettre une adaptation de la méthode au sujet développé, offrant de plus aux lecteurs la possibilité de s'attarder plus spécialement sur l'un des deux domaines sans connaissance particulière dans l'autre.

La sécurité du réseau local de l'Institut d'Informatique nous paraît suffisante au niveau des moyens disponibles, elle pourrait sans doute être améliorée par une gestion plus pointue de ces moyens et une prise de conscience commune de la part tant de l'administrateur que des utilisateurs. L'Université doit toutefois rester, nous semble-t-il, un temple de la coopération et du partage, le plus difficile dans ce contexte étant de percevoir cet équilibre dont nous avons fait plusieurs fois mention entre la communication totale et la sécurité à outrance.

La nécessité pour l'administrateur de la configuration de maîtriser les évolutions des performances du système dont il a la responsabilité implique qu'il dispose de moyens pour analyser ces performances et pour les communiquer aux décideurs et aux autres personnes concernées. L'outil de mesure développé l'a été dans ce contexte et ses qualités de correction, de convivialité et de réutilisabilité externe de ses résultats témoignent de son orientation vers le management de configuration. Ce logiciel nous aura déjà permis de tirer quelques enseignements de la configuration en mettant en évidence sa suffisance tant quantitative que qualitative. L'extension de cette configuration distribuée est donc pleinement envisageable et l'outil de mesure devrait être le garant de son évolution positive.

Au terme de ce travail, nous pouvons affirmer que la gestion tant de la sécurité que des performances d'un système informatique doit être partie intégrante du rôle de l'administrateur, voir constituer la fonction unique d'une personne dans une organisation plus étendue. Cette fonction doit être considérée comme une fonction de management au sens propre du terme. Elle doit se donner une politique basée sur une étude des besoins et mener à bien la réalisation des objectifs fixés par cette politique en utilisant les moyens disponibles. Puisse ce mémoire être le catalyseur de cette prise de conscience et aider les gestionnaires à concevoir une politique cohérente en matière de

sécurité et de performance et à disposer des outils nécessaires à l'aboutissement de leurs prévisions. L'Université doit jouer dans ce contexte un rôle primordial de terrain d'expérience et de développement, l'esprit de travail et la richesse intellectuelle qui la caractérisent en font l'exemple idéal devant bénéficier aux autres organisations.

ANNEXES

ANNEXE A GLOSSAIRE

**Veillez vous référer au
volume II de ce mémoire
pour le contenu de cette annexe**

ANNEXE B

LE PROGRAMME PWADM

**Deuillez vous référer au
volume II de ce mémoire
pour le contenu de cette annexe**

ANNEXE C

LE PROGRAMME SUW

**Deuillez vous référer au
volume II de ce mémoire
pour le contenu de cette annexe**

ANNEXE D

LES COMMANDES DE MESURE

**Veillez vous référer au
volume II de ce mémoire
pour le contenu de cette annexe**

ANNEXE E

LA COMMANDE ETHERFIND SUR SUN

**Veillez vous référer au
volume II de ce mémoire
pour le contenu de cette annexe**

ANNEXE F

CODE SOURCE DES MODULES

**Deuillez vous référer au
volume II de ce mémoire
pour le contenu de cette annexe**

ANNEXE G

LES FICHIERS WS*.DAT

**Deuillez vous référer au
volume II de ce mémoire
pour le contenu de cette annexe**

ANNEXE H

LES FICHIERS AUX*.DAT

**Deuillez vous référer au
volume II de ce mémoire
pour le contenu de cette annexe**

ANNEXE I

LES FICHIERS INIT*.DAT

**Veillez vous référer au
volume II de ce mémoire
pour le contenu de cette annexe**

ANNEXE J

LES FICHIERS COMP*.DAT

**Deuillez vous référer au
volume II de ce mémoire
pour le contenu de cette annexe**

ANNEXE K

LES FICHIERS HEURE.DAT ET DATE.DAT

**Deuillez vous référer au
volume II de ce mémoire
pour le contenu de cette annexe**

ANNEXE L

LA CRONTAB

**Veillez vous référer au
volume II de ce mémoire
pour le contenu de cette annexe**

ANNEXE M

LA MACRO-COMMANDE EXCEL

**Deuillez vous référer au
volume II de ce mémoire
pour le contenu de cette annexe**

ANNEXE N

LE FICHIER PERF.INTERPRET DE DEC

**Deuillez vous référer au
volume II de ce mémoire
pour le contenu de cette annexe**

**FACULTES
UNIVERSITAIRES
N.D. DE LA PAIX**

NAMUR

INSTITUT D'INFORMATIQUE

**LES PERFORMANCES
ET LA SECURITE
D'UN RESEAU LOCAL**

par Alain DELVAUX

et

Dany DONNEN

Erratum

Promoteur :

Professeur J. RAMAEKERS

Mémoire présenté en vue
de l'obtention du titre
de Licencié et Maître
en Informatique

Année académique 1989 - 1990

Monsieur le Président,

Monsieur le Professeur,

Messieurs,

Suite à la relecture de la version définitive de notre mémoire nous nous sommes aperçus de la présence d'erreurs dans le condensé et dans d'autres parties du texte. Aux fins de ne pas laisser plus longuement ces erreurs dans notre travail, nous nous proposons d'y annexer cet erratum.

Pour nous permettre de vous fournir une bibliographie tout à fait exacte, nous avons été dans l'obligation de ne pas l'insérer dans la version définitive du mémoire, l'origine de certaines sources ne nous étant pas encore connue . La bibliographie fait donc partie de cet erratum et doit être insérée à la fin du texte. La table des matières a été modifiée en conséquence.

Les pages à insérer ou à remplacer sont reprises en annexe de cet erratum. En plus de ces modifications nous vous demandons d'apporter les corrections suivantes :

page 59 : 2.4.1.2.3., 2ième ligne, supprimer "(voir annexe ?)"

page 76 : 3ième paragraphe, 4ième ligne, lire "en annexe C" au lieu de "en annexe ?"

page 193 : 3.5.2.2.1., 2ième paragraphe, 2ième ligne, supprimer "sont représentées sur le graphique suivant"

Abstract

Security and performances of a local area network are two essentials components of its good running.

The security of the network of the Computer Science Institut~~e~~ has been considered by using a client/server model. This approach allowsto evaluate each factor in the process of security of a system and to locate the weaknesses of some servers.

The mesure of the performances of the system has been considered as a help to configuration management. A measurement software of the evolution of the activity has been developped to constitute for the manager a decision help tool.

This two approaches have allowed to show that, both the security level and the health state of the studied system, answer to the needs of the users and the managers.

Condensé

La sécurité et les performances d'un réseau local constituent deux éléments essentiels de son fonctionnement.

La sécurité du réseau local de l'Institut d'Informatique a été envisagée selon un modèle client/serveur. Cette approche permet d'évaluer chaque facteur intervenant dans le processus de sécurisation d'un système et de localiser les failles inhérentes à certains serveurs.

La mesure des performances du système a été considérée comme une aide au management de la configuration. Un logiciel de mesure de l'évolution de l'activité a été développé, dans le but de constituer pour l'administrateur un premier outil d'aide à la décision.

Ces deux approches ont permis de montrer que tant le niveau de sécurité que l'état de santé du système étudié répondent aux besoins des utilisateurs et des administrateurs.

BIBLIOGRAPHIE

1. ASSOCIATION DES BANQUES BELGES, "Combattre la fraude informatique par la sécurité logique", Aspects et Documents 79, Août 1988.
2. AUERBACH STAFF (The), "Computer Performance Evaluation", Data Processing Management, Auerbach Publishers Inc., Boston, 1984.
3. AUERBACH STAFF (The), "Change Control Management", Data Processing Management, Auerbach Publishers Inc., Boston, s.d.
4. BARBER R.E. and LUCAS H.C.Jr, "System Response Time, Operator Productivity and Job satisfaction", Communications of the ACM, Vol 11, n0 26, Août 1983.
5. BEILNER H. (University of Dortmund), "Performance Measurements and Tools" in the 7th International Conference on Distributed Computing Systems - Berlin 21-25 september 1987, Computer Society Press of the IEEE, Washington, 1987.
6. BOI L., BOURRET P., CROS P., GENTHON P., MARTIN R., MICHEL P., TREPOS R. et ROUSSELOT J.Y., "Mesures des systèmes informatiques (A l'usage des ingénieurs systèmes)", CEPAQUES Ed., Toulon, 1978.
7. BOURNE S., "Le Système Unix", Traduit de l'américain, Interéditions, Paris, 1985.
8. BRICE R.S. and SHERMAN S.W., "Empirical Comparison of partitionned and non-partitionned Buffer Management in Virtual Memory Systems", Computer Performance Evaluation, Chamelon Press Ltd., London, 1976.

9. BUZEN J.-P., "Host/Network Response Time Planning", Data Processing Management, Auerbach Publishers Inc, Boston, 1984.
10. CARROL J.M., "Managing Risk - A Computer Aided Strategy", Butterworth Publishers Inc., Boston, 1984.
11. CHURCHWARD M., "Classification of Security in Systems International", Reed Business Publishing, March 1989, Haywards Heath, 1989.
12. "Computer Performance Evaluation", Chamelon Press Ltd., London, 1976.
13. COOPER J.A. (Sandia National Laboratories), "Computer - Security Technology", Lexington Books, Lexington, 1984.
14. CROMBE Y. et WARTON L., "La Sécurité des Systèmes Informatiques", Recherche financée par le Collège Interuniversitaire d'Etudes Doctorales dans les Sciences du Management, Université Catholique de Louvain-la-Neuve - Institut d'Administration et Gestion, 1982.
15. DETREMBLEUR B., "Network Management in an Interconnected Networking Environnement : The Current State of Art from the OSI point of vue and a Description of a Practical Implementation", Mémoire, Facultés Universitaires Notre-Dame de la Paix - Institut d'Informatique, Namur, 1987.
16. DETREMBLEUR B., "Le transfert de fichiers et le remote-login sur les protocoles DECnet et TCP-IP", Rapport technique, Facultés Universitaires Notre-Dame de la Paix - Institut d'Informatique, Namur, 1989.
17. FARQUHAR B., "Network Security : Evaluation of Risk in a telecommunication Network", Computer Fraud and Security Bulletin, Vol 11, Number 3, January 1989.
18. FERRARI D., SERAZZI G. and ZEIGNER A., "Measurement and Tuning of Computer Systems", Prentice-Hall Inc., Englewood Cliffs, 1983.

19. **FERRARI D. and BUNT R.B., "Measuring Computer System Performance", Data Processing Management, Auerbach Publishers Inc., Boston, 1985.**
20. **F.I.A.B. (3ème Congrès), "Sécurité et Protection en Informatique", F.I.A.B., Bruxelles, 1982.**
21. **FINNERAN E.D., "Security supervision - A Handbook for Supervisors and Managers", Butterworth Publishers Inc., Boston, 1981.**
22. **FISHER R.P. (IBM Corporation), "Information Systems Security", Prentice-Hall Inc., Englewood Cliffs, 1984.**
23. **GASSER M. (Research Director - Digital Equipment Laboratory), "Security in Distributed Systems", draft, Conference on Network Technology, IBM Chair, Leuven, March 1990.**
24. **GRIESMER S., "An Overview of UNIX", Data Processing Management, Auerbach Publishers Inc., Boston, 1986.**
25. **GUYNES J.L., "Impact of System Response Time on State Anxiety", Communications of the ACM, Vol 31, Number 3, March 1988.**
26. **HAMMOND J.L. and O'REILLY P.J.-P., "Performance Analysis of Local Computer Networks", Addison-Wesley Publishing Company, Massachusetts, 1986.**
27. **HSIAO D.K., KERPS D.S. and MADNICK S.E., "Computer Security", ACM Monograph Series, Academic Press Inc., New-York, 1979.**
28. **KATZAN M.Jr, "Computer Data Security", Computer Science Series, 1973.**

29. KINDERLEHRER R., **"Handbook for Data Center Management"**, Q.E.D., Information Science Inc., Wellesley, 1979.
30. LAMERE J.-M., **"La Sécurité Informatique - Approche Méthodologique"**, Dunod Informatique, Paris, 1985.
31. LEFFER, McKUSICK, KARELS and QUATERMAN, **"Design and Implementation of the Unix 4.3 BSD Operating System"**, Addison-Wesley Publishing Company, Massachusetts, 1987.
32. MARTIN J., **"Security Accuracy and Privacy in Computer Systems"**, Series in Automatic Computation, Prentice-Hall Inc, Englewood Cliffs, 1973.
33. OLIVER E. and WILSON J., **"Practical Security in Commerce and Industry"**, Fourth Edition, Gower Publishing Company Ltd., Aldershot, 1983.
34. PARKER D.B., **"Crime by Computer"**, Charles Scribner's sons, New-York, 1976.
35. POUNDER CNM Dr, KOSTEN M., PAPADOPOULOS S. and RICHARD A., **"Managing Data Protection"**, CIPFA/CCS, Chartered Institute of Public Finance and Accountancy, Chameleon Press Ltd., London, 1987.
36. PRATA S. and MARTIN D., **"Unix System V Bible - Commands and Utilities"**, Hayden-Books, Indianapolis, 1987.
37. QUISQUATER J.-J. , **"Les techniques cryptographiques"**, Note de cours - séminaire organisé dans le cadre du cours de Sécurité des Systèmes Informatiques du Professeur RAMAEKERS, Facultés Universitaires Notre-Dame de la Paix - Institut d'Informatique, Namur, 1989.

38. RAMAEKERS J., "La Sécurité des Systèmes Informatiques", Note de Cours de 2ème Licence et Maîtrise en Informatique, Facultés Universitaires Notre-Dame de la Paix - Institut d'Informatique, Namur, 1990.
39. RAVINDER , "Local Area Network : The Network Manager's Perspective", Information and Management, Elsevier Science Publishers, North-Holland, 1988.
40. RIFFLET J.-M., "La programmation sous Unix", Mc Graw-Hill, Paris, 1986.
41. RITCHIE D.M. and THOMPSON K., "The Unix Time-Sharing System", revised version of article appeared in the Communications of the ACM, Vol 17, number 7, July 1974.
42. RUSHINEK A. and RUSHINEK S., "What makes users happy?", Communications of the ACM, Vol 29, number 7, July 1986.
43. SABOUREAU J.-P., "Contrôle des Performances de l'Exploitation d'un Centre Informatique", Editions Hommes et Techniques, Boulogne-Billancourt, 1981.
44. SAGE R.G., " Tricks of the Unix Masters", The Wait Group Inc, Indianapolis, 1987.
45. SAUNDERS M., "Protecting your Business Secrets", Grower, Cambridge, 1985.
46. SINGLETON J.-P., McLEAN E.R. and ALTMAN E.N., "Measuring Information Systems Performance : Experience with the Management by Results Systems at Security Pacific Bank", MIS Quaterly, Vol 12, Number 2, June 1988.
47. SLONE J.P., "Assessing LAN Technologies", Data Processing Management, Auerbach Publishers Inc., Boston, 1976.

48. TALBOT J.R., "Management Guide to Computer Security", Grower, Cambridge, 1981, Reprinted in 1985.
49. VAN LAMSWEERDE A. , "Méthodologie de développement de Logiciel", Notes de cours de 2ème Licence et Maîtrise en Informatique, Facultés Universitaires Notre-Dame de la Paix - Institut d'Informatique, Namur, 1989.
50. VAN TASSEL D. (University of California, Santa-Cruz), "Computer Security Management", Prentice-Hall Inc., Englewood Cliffs, 1972.
51. WOOD PH. and KOCHAN S.G., "Unix system Security", Hayden-Books, Indianapolis, 1985.

3.5. UTILISATION DE L'OUTIL.....	188
3.5.1 PERTINENCE DES VALEURS ET MESURES.....	188
3.5.1.1. CORRECTION DES MESURES.....	188
3.5.1.2. PERTINENCE DES MESURES.....	191
3.5.2. REALISATION DE L'ANALYSE.....	191
3.5.2.1. HYPOTHESES.....	191
3.5.2.2. ANALYSE DES MESURES COLLECTEES.....	193
3.5.2.2.1. Utilisation du CPU.....	193
3.5.2.2.2. Accès au disque.....	199
3.5.2.2.3. Occupation de la mémoire	202
3.5.2.2.4. Activité de pagination.....	204
3.5.2.2.5. Contribution au trafic du réseau.....	206
3.5.2.3. ENSEIGNEMENTS.....	211
3.6. CONCLUSIONS.....	213

CONCLUSIONS	215
--------------------------	------------

BIBLIOGRAPHIE	218
----------------------------	------------

ANNEXES

ANNEXE A.....	GLOSSAIRE
ANNEXE B.....	PROGRAMME PWADM
ANNEXE C.....	PROGRAMME SUW
ANNEXE D.....	COMMANDES UNIX DE MESURES
ANNEXE E.....	COMMANDE ETHERFIND SUR SUN
ANNEXE F.....	CODE SOURCE DU MESUREUR
ANNEXE G.....	FICHIERS WS*.DAT
ANNEXE H.....	FICHIERS AUX*.DAT
ANNEXE I.....	FICHIERS INIT*.DAT
ANNEXE J.....	FICHIERS COMP*.DAT
ANNEXE K.....	FICHER HEURE.DAT
ANNEXE L.....	CRONTAB
ANNEXE M.....	MACRO-COMMANDE EXCEL
ANNEXE N.....	FICHER PERF.INTERPRET DE DEC

**FACULTES
UNIVERSITAIRES
N.D. DE LA PAIX**

NAMUR

INSTITUT D'INFORMATIQUE

**LES PERFORMANCES
ET LA SECURITE
D'UN RESEAU LOCAL**

**par Alain DELVAUX
et
Dany DONNEN**

**Promoteur :
Professeur J. RAMAEKERS**

**Mémoire présenté en vue
de l'obtention du titre
de Licencié et Maître
en Informatique**

Année académique 1989 - 1990

ANNEXE A GLOSSAIRE

authentification : vérification par le système que le nom du sujet effectuant une requête lui appartient bien (par exemple grâce à un mot de passe)

awk : commande de manipulation de documents en Unix

backbone : épine dorsale du support de transmission d'un réseau

bombe logique : technique d'attaque de la sécurité informatique également appelée bombe à retardement, c'est un programme dont l'exécution est déclenchée par la réalisation d'un événement particulier, son principe est similaire à celui du virus mais ne comprend aucune reproduction

bridge : système qui permet de répéter les signaux émis sur un support de transmission d'un segment à un autre, il peut parfois réaliser en plus une sélection des signaux à répéter

bus : topologie de réseau local du type distribué : les machines sont connectées sur un même support et lorsque l'une d'elles émet, toutes entendent

câble coaxial : support de transmission constitué de deux conducteurs concentriques, c'est ce moyen qui est utilisé pour véhiculer les signaux de télévision

cheval de Troie : technique d'attaque de la sécurité informatique utilisant un programme qui simule

l'exécution d'une commande ou d'un processus connu de la victime, tout en y ajoutant l'exécution d'une ou plusieurs autres commandes qui mettent en péril la sécurité de celui qui l'a activé ou d'une tierce personne

chiffrement : transformation de l'information en chiffres au moyen d'un algorithme mathématique particulier de façon à la rendre illisible

clé de chiffrement : secret nécessaire au chiffrement ou au déchiffrement de l'information

cluster : groupement de machines qui permet à l'utilisateur de les voir comme une seule.

commutation : principe très répandu de communication par réseau dont l'idée est qu'une connexion temporaire est établie entre les deux parties durant le temps de la communication

commutation par paquets : système le plus fréquemment utilisé en communication de données, le message est décomposé en séquences de petits paquets qui sont envoyés indépendamment à travers le réseau

configuration management : administration d'une configuration informatique incluant l'étude de ses performances et de ses possibilités d'extension

csma/cd : le plus courant des système de réseau locaux, l'accès multiple y est géré par le principe de l'écoute avant l'émission (Carrier Sense), en cas de détection de collision (Collision Detection) une réémission sera effectuée après un certain temps

dcs : réseau public belge de commutation par paquet (Data Commutation Service) auquel on accède grâce au protocole X25

decnet : protocole de communication appartenant à Digital Equipment Corp.

delni : appareil comparable à un gros domino permettant l'interconnexion de différentes machines

démon : processus exécuté périodiquement par le système pour assurer son rôle de gestion

dempr : appareil assurant l'interconnexion entre un câble et plusieurs câbles plus fins

des : algorithme standard de chiffrement des données proposé par le National Bureau of Standard

decrypt : outil de chiffrement Unix basé sur le DES, il chiffre l'entrée standard au moyen d'une clé chiffrée à l'aide de la commande crypt de Unix appliquée à la clé donnée en argument

desta : appareil similaire à un transceiver et permettant aussi l'interconnexion de différentes machines

entrée par l'arrière : technique d'attaque de la sécurité informatique par laquelle un accès, pourtant connu du concepteur, est découvert et utilisé par un attaquant, par exemple pour obtenir un accès illégitime à un équipement ou à des données

etherfind : commande du système d'exploitation des stations de travail Sun qui donne une série de renseignements sur les accès au réseau

ethernet : réseau local développé par Xerox du type CSMA/CD qui a conduit à un standard officiel (IEEE 802.3)

event-driven : méthode de prise de mesures dont le déclenchement est provoqué par un événement système

fibre optique : support de transmission très performant comparable à un miroir cylindrique dans lequel la lumière se propage par réflexion

holter monitoring : système d'enregistrement magnétique des pulsations cardiaques d'un patient durant une période généralement de 24 heures, dans le but d'analyser l'évolution et les événements exceptionnels qui se seraient produits

hôte : ordinateur possédant son propre espace de fichier et caractérisé par un nom, un système d'exploitation et un protocole

i-node : (index-node) noeud de l'arbre qui représente la hiérarchie des fichiers et des répertoires en Unix

identification : association d'un sujet à un nom ou à une classe (par exemple au moyen d'un nom d'utilisateur)

inférence : menace passive en sécurité informatique, elle est basée sur la déduction d'informations utiles à partir des caractéristiques de données protégées contre l'observation directe

insertion : technique d'attaque active contre la sécurité informatique qui consiste en l'ajout d'informations par une personne non-autorisée, tout en faisant croire que cette information a été écrite par son auteur légitime

intrusion ("hacking") : nom communément donné à la technique qui consiste à deviner par essais successifs la clé d'authentification d'un utilisateur légitime d'un système

kermit : logiciel permettant la communication entre machines hétérogènes (PC, Macintosh, Vax VMS, ...)

lan : ou réseau local (Local Area Network) est un système de communication distribué : toutes les machines connectées écoutent et il n'y a pas de routage

load average : taille moyenne de la file des processus à l'entrée d'un processeur

macro-commande : commande personnalisée qui permet l'exécution d'une série de commandes de base d'un logiciel

mascarade : technique d'attaque d'attaque active contre la sécurité informatique ayant pour but, pour une personne non-autorisée, de s'identifier à une autre personne

m o d e m : le modulateur-démodulateur est un appareil qui permet l'émission et la réception d'informations binaires sur le réseau téléphonique analogique

modification : menace active en sécurité informatique par laquelle l'attaquant modifie des informations dont l'accès lui est interdit

observation : principale menace passive en sécurité informatique, elle consiste en l'acquisition de l'information par une personne non-autorisée

overhead : données véhiculées sur le réseau pour des raisons internes de fonctionnement

pabx : auto-commutateur privé qui réalise la commutation et le transfert de signaux téléphoniques et digitaux, il joue un rôle comparable à celui d'une centrale téléphonique.

paire torsadée : support de transmission constitué de deux fils enroulés en spirale, c'est ce moyen qui est utilisé pour les communications téléphoniques

perms : programme de vérification du propriétaire et des permissions d'accès des fichiers en Unix

processus shell : processus résultant de l'exécution d'un programme shell

programme shell : programme écrit dans le langage de commande de Unix

protocole : ensemble de règles permettant à deux machines de se comprendre et de communiquer.

pwadm : programme qui facilite la gestion de la durée de validité imposable par l'administrateur aux mots de passe des utilisateurs en Unix

raccourci-clavier : combinaison de touches permettant le déclenchement d'une commande sans utiliser le menu

refus de service : large classe de menaces en sécurité informatique qui se rapportent plus souvent au problème de la fiabilité du système qu'à la protection des données

remote login : connexion à un ordinateur à partir d'une autre via le réseau

remote repeater : système réalisant l'interconnexion de deux segments de câble, il permet de répéter les signaux émis sur un support de transmission d'un segment à un autre

remote shell : commande shell exécutée à partir d'une machine sur une autre via le réseau

répétition : technique d'attaque contre la sécurité informatique qui permet à son auteur d'insérer une

information identique à celle observée précédemment

sampling : méthode de prise de mesures à périodicité constante qui apporte un échantillon suffisant pour la détermination des valeurs réelles

secure : outil d'audit de la sécurité permettant à l'administrateur d'exécuter une série de vérifications sur le système et de mesurer son degré de sécurité

segment : partie constitutive d'un support de transmission, c'est un câble dont la longueur est limitée à 500 m pour un réseau Ethernet

serveur : machine qui fournit et qui gère des ressources partagées par d'autres (disque, imprimante, ...).

serveur de terminal : appareil qui permet l'accès au réseau à partir d'un terminal.

shell : langage de commande qui réalise l'interface entre l'utilisateur et le système Unix

station de travail : ordinateur individuel faisant partie d'un système distribué et répondant à certaines caractéristiques techniques.

suw : programme qui analyse le fichier Unix *sulog* contenant des informations sur toutes les transactions effectuées avec la commande *su* et imprime les noms de tous les utilisateurs qui se sont illégalement substitués à *root*.

système de fichier : entité de gestion de fichiers

tcp/ip : le Transmission Control Protocole / Internet Protocole est un exemple très répandu de protocole de communication en réseau local.

terminateur : appareil qui doit être placé à chaque extrémité d'un segment pour maintenir une tension constante sur le câble

transceiver : appareil assurant le rôle de prise de connexion indispensable à tout branchement au câble.

udp/ip : protocole de communication non-orienté connexion

us-dod : département de la défense du gouvernement américain

uucp : (Unix to Unix CoPy) programme Unix qui permet la copie de fichiers entre machines distantes

worm ("worm") : technique d'attaque de la sécurité informatique basée sur un processus qui se déplace constamment d'un endroit de la mémoire à un autre ou d'une machine à une autre via les moyens de communication et scanne les différentes zones de la mémoire dans le but de trouver les informations pertinentes

virus : technique d'attaque de la sécurité informatique qui a pour effet d'infecter les programmes ou les données du système en les modifiant de façon arbitraire, de telle sorte que ceux-ci véhiculent à leur tour des virus

vol de temps informatique consiste en l'utilisation sans

autorisation de la capacité informatique d'autrui à des fins personnelles

wiretapping : technique qui permet de se connecter de façon pirate sur un support de transmission de données, soit par connexion physique directe, soit grâce à une captation par induction

x25 : protocole d'accès au réseau public DCS de la RTT destiné aux transmissions de données digitales par commutation de paquets.

x400 : protocole largement répandu dans le courrier électronique

ANNEXE B

LE PROGRAMME PWADM

```

/*****
/*   password administration program   */
*****/

#include <stdio.h>
#include <signal.h>
#include <pwd.h>
#include <ctype.h>
#include <sys/types.h>
#include <time.h>

#define TRUE      1
#define FALSE     0
#define SECSPERWEEK (60 * 60 * 24 * 7)

main (argc,argv)
int argc;
char *argv[];
{
    int                error = FALSE, change = FALSE, disable =
                        FALSE, nochange = FALSE, print = FALSE,
                        chrequest = FALSE, min = -1, max = -1, found,
                        option;

    long               aging, saveaging, a641 (), time(), now,
                        pwmax = -1, pwmin = -1;

    struct passwd      *pwptr, *pwptr2, *getpwnam (), *getpwent (),
                        *getpwuid ();

    char               username [L_cuserid], *164a ();

    static char        tmpfile[20] = "/ect/pass", pwdtmp[] =
                        "/etc/ptmp", oldpasswd[] = "/etc/opasswd",
                        agebuf[20];

    time_t             lastchange = 0;

    FILE               *tmp;

    extern int         optind;

    extern char        *optarg;

```

```

/* process option */

while ((option = getopt (argc, argv, "cdm:npM:")) != EOF)
    switch (option) {
        case 'c':
            change = TRUE;
            break;

        case 'd':
            disable = TRUE;
            break;

        case 'm':
            if (sscanf(optarg, "%d", &min) != 1 || min < 0 )
                {fprintf(stderr, "badminimum value.\n");
                 exit (1);
                }
            break;

        case 'M':
            if (sscanf(optarg, "%d", &max) != 1 || max < 0 )
                {fprintf(stderr, "badmaximum value.\n");
                 exit (1);
                }
            break;

        case 'n':
            nochange = TRUE;
            break;

        case 'p':
            print = TRUE;
            break;

        case '?':
            error = TRUE;
    }

if (min + max != -2 || change || nochange || disable )
    chrequest = TRUE;

```



```

    exit (2);
}

if (*pwptr -> pw_age != '\0')
{
    aging = a641(pwptr->pw_age);
    saveaging = aging;
    pwmax = aging & 077;
    pwmin = (aging >> 6) & 077;
    lastchange = (time_t) (aging >> 12);
}

if (chrequest)
{
    if (min != -1) pwmin = min;
    if (max != -1) pwmax = max;
    if (pwmin != -1 && pwmax == -1) pwmax = 0;
    else if (pwmax != -1 && pwmin == -1) pwmin = 0;
    if (change) {lastchange = 0;
    if (pwmin == -1 && pwmax == -1 || pwmin > pwmax) pwmin = pwmax = 0;
    }
}

else
    if (nochange)
    {
        lastchange = 0;
        pwmin = 1;
        pwmax = 0;
    }

    if (disable) pwmin = pwmax = 0;
    aging = pwmax + (pwmin << 6) + (lastchange << 12);
    pwptr->pw_age = 164a (aging);
    if (strlen(pwptr->pw_age) == 1 ) strcat (pwptr->pw_age, ".");

/* min is zero */

    if (change) strcat (pwptr->pw_age, "..");
}

stropy (agebuf, pwptr->pw_age);

```

```

now = time ((long *)0)/secsperweek;
if (print)
    {if (disable || pwmin > pwmax || pwmin == 0 && pwmax == 0 ||
        pwmin == -1 && pwmax == -1)
        {printf ("password aging not in effect\n");
        if (agebuf[0] == '.' && agebuf[1] == '.')
            printf ("but password must be change on next login\n");
        else if (pwmin > pwmax) printf ("and password can't be changed\n");
        }
    else
        {
            if (lastchange > now || now > lastchange + pwmax)
                {
                    printf("password expired\n");
                    printf ("password must be changed every %ld weeks\n", pwmax);
                    printf ("once changed, it can not be changed again for ");
                    printf ("another %ld weeks\n", pwmin);
                }
        }
    if (!chrequest) exit(0);

/* create temporary file */

umask (0177);
sprintf (tmpfile + strlen(tmpfile), "%d", getpid());
if ((tmp = fopen(tmpfile, "w")) == NULL)
    {
        fprintf (stderr, "can't access temp file, no changes made\n");
        exit(3);
    }
signal (SIGHUP, SIG_IGN);
signal (SIGINT, SIG_IGN);
signal (SIGQUIT, SIG_IGN);

/* link to temporary file used by passwd, pwadm, and maybe others */

if (ln(tmpfile, pwtmp)) {
    rm (tmpfile);

```

```

        exit (3);
    }

/* copy over password entries, updating matching one */

found = FALSE;
while ((pwptr2 = getpwent()) != (struct passwd *) NULL)
    {
        if (!found && strcmp (pwptr2->pw_name, username) == 0)
            {
                pwptr2->pw_age = agebuf;
                found = 2;
            }
        putpwent (pwptr2, tmp);
    }
endpwent();
fclose(tmp);
if (mv("/etc/passwd", oldpasswd))
    {
        unlink (tmpfile);
        unlink (pwdtmp);
        exit(3);
    }
if (ln(pwdtmp, "/etc/passwd"))
    {
        if (mv(oldpasswd, "/etc/passwd")) fprintf (stderr, "can't
        restore /etc/passwd !\n");
        unlink (tmpfile);
        unlink (pwdtmp);
        exit(3);
    }
chmod ("/etc/passwd", 0444);
unlink (tmpfile);
unlink (pwdtmp);
exit(0);
}
ln (from, to)
char *from, *to;

```

```

{if (link(from, to) == -1)
    {
        fprintf (stderr, "can't link %s to %s \n", from, to);
        fprintf (stderr, "password file not updated\n");
        return (-1);
    }
return (0);
}

```

```

rm (file)
char *file;
{if (access (file, 0) == 0 && unlink(file) == -1)
    {
        fprintf (stderr, "can't unlink %s\n", file);
        fprintf (stderr, "password file not updated\n");
        return (-1);
    }
return (0);
}
mv (from, to)
char *from, *to;
return(rm(to) || ln(from, to) || rm(from));
}

```


ANNEXE C

LE PROGRAMME SUW

static char id []="@(#)suw v1.0

Author: Russ Sage";

#include <stdio.h>

#define FALSE 0

define TRUE 1

define MATCH 0

define BSIZ 80

main(argc,argv)

int argc;

char *argv [];

{

register int alert, c, mail, n;

FILE *fp1, *fp2;

char *p, *uname, line[BSIZ], tmp[BISZ],

*log = "/usr/adm/sulog";

static char *legal[] = {"sage-root\n", "roor-root\n", NULL}...

static char *adm[] = {"sage", NULL};

mail =FALSE;

if (argc > 1 && argv[1][0] == '-')

switch (argv[1][1])

{

case 'm':

mail = TRUE;

--argc;

++argv;

break;

default:

fprintf(stderr,"suw: invalid argument %s\n", argv[1])

fprintf(stderr,"usage: suw [-m] [sulog]\n");

exit(1);

}

```

if (argc == 2)
    log = *++argv;

if ((fp1 = fopen(log,"r")) == NULL)
{
    fprintf(stderr,"suw : error opening %s\n", log);
    fprintf(stderr,"usage: suw [-m] [sulog]\n");
    exit(1);
}

sprintf(tmp, "/tmp/suw%d", getpid());
if ((fp2 = fopen(tmp,"w+")) == NULL)
{
    fprintf(stderr,"suw : error opening %s\n", tmp);
    fprintf(stderr,"usage: suw [-m] [sulog]\n");
    exit(1);
}

while (fgets(line,sizeof(line),fp1) != NULL)
{
    p = line + 15;
    if (*p == '+')
    {
        p = p+2;
        while (*p != ' ') p++;
        p++;
        uname = p;
        while (*p && *p++ != '-') continue;
        if (strcmp(p, "root\n") == MATCH)
        {
            alert = TRUE;
            for (n=0; legal[n] != NULL; n++)
                if (strcmp (uname,legal[n]) == MATCH)
                {
                    alert = FALSE;
                    break;
                }
        }
    }
}

```

```

        if (alert) fprintf(fp2,"Illegal --> %s", line);
    }
}

if (mail)
{
    fclose (fp2);
    for (n=0; adm[n] != NULL; n++)
    {
        sprintf(line, "cat%s | mail %s", tmp, adm[n]);
        system(line);
    }
}
else
{
    rewind(fp2);
    while ((c = getc(fp2)) != EOF)
        putc(c,stdout);
    fclose(fp2);
}

fclose (fp1);
unlink(tmp);
}

```

ANNEXE D

LES COMMANDES DE MESURE

D.1. COMMANDE DF

D.1.1. MANUEL

NAME

df - display free and used disk space

SYNTAX

df [-i] [-n] [file system...] [file...]

DESCRIPTION

The `df` command displays the amount of disk space available on the specified `file system`, for example, `/dev/r0a`. It also displays the amount of available disk space on the file system in which the specified `file` is contained, for example, `$HOME`. If a device is given that has no file systems mounted on it, `df` displays the information for the root file system. Without any arguments or options, `df` displays both the used disk space and the free disk space on all of the file systems that have been mounted using the `fstab` file. The numbers are reported in kilobytes. Unless the `-n` option is specified, `df` updates the statistics stored in memory for the file system specified, before it returns the information.

OPTIONS

- i Also report the number of used and free inodes.

- n Do not update the file system statistics stored in memory. Instead, returned whatever statistics are stored in memory. This prevents `df` from hanging in the event that a server containing the specified file system is down.

RESTRICTIONS

You cannot use the `df` command to find free space on an unmounted file system using the block or character special device name. Instead, use the `dumpfs` command.

EXAMPLES

```
% df
Filesystem      Total    kbytes kbytes  %
node            kbytes  used   free   used  Mounted on
/dev/ra1a       7429    2085   4602   31%   /tmp
/dev/ra0e       30519   14817  12651   54%   /usr/spool
/dev/ra0h       313233  122858 159052   44%   /usr/staff1
```

The total disk space is the total space that was created during the making of the file system. The addition of the used space, the free space and a percentage of reserved space is the total space. The default value for the reserved space is 10%.

FILES

/etc/fstab List of mounted file systems

SEE ALSO

getmnt(2), fstab(5), dumpfs(8), icheck(8), mkfs(8), newfs(8), quot(8)

D.1.2. RESULTAT

Filesystem	Total	kbytes	kbytes	%	
node	kbytes	used	free	used	Mounted on
/dev/ra0a	7423	6309	372	94%	/
/dev/ra0e	187927	159249	9886	94%	/usr
/dev/ra0f	14383	2477	10468	19%	/var
/dev/ra0d	14383	9	12936	0%	/tmp
/dev/ra1g	177071	147466	11898	93%	/users
/dev/ra1a	28799	19767	6153	76%	/var/diskless/dlenv0
/dev/ra1b	162099	133266	12624	91%	/var/diskless/dlclient0
/dev/ra0h	96399	39006	47754	45%	/users/students

D.2. COMMANDE IOSTAT

D.2.1. MANUEL

NAME

iostat - report I/O statistics

SYNTAX

iostat [-i -n -t -e -r -v -a -l[-c -o -u -n -t]]

DESCRIPTION

The iostat command iteratively reports the number of characters read and written to terminals. For each disk, it reports the number of seeks, transfers per second, and kilo-bytes transferred per second. Additionally, it provides the percentage of time the system has spent in user mode, in user mode running low priority (niced) processes, in systemmode, and idling.

To compute this information seeks, data transfer completions and number of words transferred are counted for each disk; for terminals collectively, the number of input and output characters are counted. Also, each sixtieth of a second, the state of each disk is examined and a tally is made if the disk is active. From these numbers and given the transfer rates of the devices it is possible to determine average seek times for each device.

The optional -i -n -t -e -r -v -a -l argument causes -i -o -s -t -a -t to report once each -i -n -t -e -r -v -a -l seconds. The first report is for all time since a reboot and each subsequent report is for the last interval only.

The optional -c -o -u -n -t argument restricts the number of reports.

FILES

/dev/kmem
/vmunix

SEE ALSO

vmstat(1)

D.2.2. RESULTAT

D.2.2.1. Iostat 1 10

tty		ra0		ra1		cpu			
tin	tout	bps	tps	bps	tps	us	ni	sy	id
1	14	7	1	12	3	15	0	40	45
0	0	0	0	0	0	41	0	59	0
0	0	0	0	0	0	29	0	71	0
0	0	0	0	4	1	35	0	65	0
0	0	0	0	0	0	33	0	68	0
0	0	0	0	0	0	26	0	74	0
0	0	0	0	0	0	30	0	70	0
0	0	0	0	0	0	22	0	78	0
0	0	0	0	0	0	29	0	71	0
0	0	0	0	0	0	24	0	76	0

D.3. COMMANDE NETSTAT

D.3.1. MANUEL

NAME

netstat - show network status

SYNTAX

```
netstat [-Aan][f address family][system][core]
netstat [himnrs][f address family][system]
[core]
netstat [n][l interface] interval[system][core]
```

DESCRIPTION

The `netstat` command displays the contents of network related data. The information presented, there are a number of output formats. The first form of the command displays a list of active sockets for each protocol. The second form presents the contents of one of the other network data structures according to the option selected. The third form, with an `interval` specified, continuously displays the information

regarding packet traffic on the configured network interfaces. If no options are specified, `netstat` displays the state of all active sockets from those using any of the protocols listed in `/etc/protocols`.

The arguments, `system` and `core` allow substitutes for the defaults `/vminix` and `/dev/kmem`. If an `interval` is specified, `netstat` display the information regarding packet traffic on the configured network interfaces continuously, pausing `interval` seconds before refreshing the screen.

There are a number of display formats, depending on the information presented. The default display, for active sockets, shows the local and remote addresses, send and receive queue sizes (in bytes), protocol, and, optionally, the internal state of the protocol.

Address formats are either of the form `host.port` or `network.port`, if a socket's address specifies a network but no specific host address. When known, the host and network addresses are displayed symbolically according to the databases `/etc/hosts` and `/etc/networks`, respectively. If a symbolic name for an address is unknown, or if the `n` option is specified, the address is printed in the Internet dot format. Refer to `inet(3n)` for more information regarding this format. Unspecified, or wildcard, addresses and ports appear as an asterisk (*).

The interface display provides a table of cumulative statistics regarding packets transferred, errors, and collisions. The network address (currently Internet specific) of the interface and the maximum transmission unit (mtu) are also displayed.

The routing table display indicates the available routes and their status. Each route consists of a destination host or network and a gateway to use in forwarding packets. The flags field shows the state of the route (for example, U if up), whether the route is to a gateway (G), and whether the route was created dynamically by a redirect (D). Direct routes are created for each interface attached to the local host. The gateway field for such entries shows the address of the outgoing interface. The `refcnt` field gives the current number of active uses of the route. Connection oriented

protocols normally hold on to a single route for the duration of a connection, while connectionless protocols obtain a route while sending to the same destination. The use field provides a count of the number of packets sent using that route. The interface entry indicates the network interface utilized for the route.

When `netstat` is invoked with an `interface` argument, it displays a running count of statistics related to network interfaces. This display consists of a column for the primary interface (the first interface found during autoconfiguration), and a column summarizing information for all interfaces. The primary interface may be replaced with another interface with the `I` option. The first line of each screen of information contains a summary since the system was last rebooted. Subsequent lines of output show values accumulated over the preceding interval.

OPTIONS

A Displays the address of any associated protocol control blocks; used for debugging.

a Displays the information for all sockets. Normally sockets used by server processes are not shown.

address_family

Limits statistics or address control block reports to those of the specified `address_family`. Recognized address families are `inet`, for AF_INET, and `unix`, for AF_UNIX.

h Displays the state of the IMP host table.

interface

Shows information only about this interface. Used with an `interface` displayed below.

i Displays status information for autocofigured interfaces. Interfaces statically configured into a system, but not located at boot time are not shown.

- m Displays information for the memory management routines
The network manages a private share of memory.
- n Displays network addresses as numbers. Normally n e t s
t a t interprets addresses and attempts to display them
symbolically.
- r Displays the routing tables. When s is also present, shows
routing statistics instead.
- s Displays per protocol statistics.
- t Displays time until interface watchdog routine starts up
(used only in conjunction with i option).

SEE ALSO

iostat(1), vmstat(1), hosts(5), networks(5), protocols(5), services(5),
trpt(8c)

D.3.2. RESULTAT

D.3.2.1. NETSTAT -M

73 mbufs (data) allocated
85 socket structures allocated
112 protocol control blocks allocated
2 routing table entries allocated
3 interface addresses allocated
2 socket names and addresses allocated
4 DECnet opts allocated
2074 RPC structures allocated
294 Kbytes allocated to network

D.3.2.2. NETSTAT -I

Name	Mtu	Network	Address	lpkts	lerrs	Opkts	Oerrs	Coll
qe0	1500	DECnet	ELEC	11484791	1	11985595	2	12633
qe0	1500	138.48	electre	11484791	1	11985595	2	12633
lo0	1536	loop	localhost	938561	0	938561	0	0

C.3.2.3. NETSTAT -S

udp:

13433267 total udp requests
0 incomplete headers
0 bad data length fields
0 bad checksums
5415 total input dropped

tcp:

410145 packets sent
245643 data packets (54912952 bytes)
23 data packets (5107 bytes) retransmitted
186056 ack-only packets (106038 delayed)
54 URG only packets
144 window probe packets
3747 window update packets
57948 control packets
453981 packets received
263078 acks (for 54969655 bytes)
50268 duplicate acks
0 acks for unsent data
203301 packets (12390690 bytes) received in-sequence
1400 completely duplicate packets (1385 bytes)
0 packets with some dup. data (0 bytes duped)
19627 out-of-order packets (2085 bytes)
0 packets (0 bytes) of data after window
10234 window probes
4619 window update packets
38 packets received after close
0 discarded for bad checksums
0 discarded for bad header offset fields
0 discarded because packet too short
19417 connection requests
19991 connection accepts
39359 connections established (including accepts)
40147 connections closed (including 26 drops)
530 embryonic connections dropped
262386 segments updated rtt (of 282323 attempts)

24 retransmit timeouts
0 connections dropped by rexmit timeout
12 persist timeouts
26287 keepalive timeouts
4698 keepalive probes sent
0 connections dropped by keepalive

icmp:

1840 calls to icmp_error
0 errors not generated 'cuz old message was icmp
Output histogram:
destination unreachable: 1840
0 messages with bad code fields
0 messages < minimum length
0 bad checksums
0 messages with bad length
Input histogram:
destination unreachable: 5
0 message responses generated

ip:

12060637 total packets received
175 bad header checksums
0 with size smaller than minimum
0 with data size < data length
0 with header length < data size
0 with data length < header length
1283051 fragments received
0 fragments dropped (dup or out of space)
33 fragments dropped after timeout
0 packets forwarded
36 packets not forwardable
0 redirects sent

D.4. COMMANDE VMSTAT

D.4.1. MANUEL

NAME

vmstat - report virtual memory statistics

SYNTAX

```
vmstat [ interval [ count ] ]  
vmstat -f  
vmstat -s [ interval [ count ] ]  
vmstat -k namelist [ corefile ]  
vmstat -z
```

DESCRIPTION

The `vmstat` command reports statistics kept about processes, virtual memory, disk, trap and cpu activity. If given a `-f` argument, it reports on the number of `forks` and `vforks` since system startup and the number of pages of virtual memory involved in each kind of fork. If given a `-s` argument, it prints the contents of the `sum` structure, giving the total number of several kinds of paging related events that have occurred since boot. If given a `-z` argument and if the UID indicates root privilege, the `sum` structure is zeroed out. If given a `-k` argument and a `namelist` and `corefile`, a dump may be interrogated to print the contents of the `sum` structure (default).

If none of these options is given, `vmstat` summarizes the virtual memory activity since the system was booted. If `interval` is specified, then successive lines are summaries of activity over the last `interval` seconds. The command `vmstat 5`, for example, prints what the system is doing every five seconds. This is a good choice of printing interval since some of the statistics are sampled in the system every 5 seconds; others vary every second. If a `count` is given, the statistics are repeated `count` times.

When you run `vmstat` the format fields are as follows:

Procs: information about numbers of processes in various states.

r in run queue
b blocked for resources (i/o, paging, etc.)
w runnable or short sleeper (< 20 secs) but swapped

Memory: information about the use of virtual and real memory. Virtual pages are considered active if they belong to processes which are running or have run in the last 20 seconds. A page is 1024 bytes.

avm active virtual pages
fre size of the free list

If the number of pages exceeds 9999, it is shown in a scaled representation. The suffix k indicates multiplication by 1000 and the suffix m indicates multiplication by 1000000. For example, the value 12345 appears as 12k.

Page: information about page faults and paging activity. These are averaged every five seconds, and given in units per second.

re page reclaims (simulating reference bits)
at pages attached (found in free list not swapdev or filesystem)
pi pages paged in
po pages paged out
fr pages freed per second
de anticipated short term memory shortfall
sr pages scanned by clock algorithm, per-second

Disk: up/hp/rk: Disk operations per second (this field is system dependent). Typically paging is split across several of the available drives. The number under each of these is the unit number.

Faults: trap/interrupt rate averages per second over the last 5 seconds.

in (non clock) device interrupts per second
sy system calls per second
cs cpu context switch rate (switches/sec)

Cpu: breakdown of percentage usage of CPU time

us user time for normal and low priority processes

sy system time

id cpu idle

If given a -S argument, the page reclaim(re) pages attached (at) fields are replaced with:

si processes swapped in

so processes swapped out

OPTIONS

-f Displays number of forks and vforks since system startup and number of pages of virtual memory involved in each kind of fork.

-s Displays total number of paging-related events occurring since boot.

-S Replaces the page reclaim (re) and pages attached (at) fields with processes swapped in (si) and processes swapped out (so).

-k Allows a dump to be interrogated to print the contents of the sum structure (default). For v m s t a t to print the contents, however, the n a m e l i s t and c o r e f i l e arguments must be specified also.

-z Zeroes out the sum structure. The user's UID must indicate root privilege.

FILES

/dev/kmem

/vmunix

D.4.2. RESULTAT

D.4.2.1. VMSTAT 1 10

procs	memory	page	disk	faults	cpu		
r b w	avm	fre	re at	pi po	fr de sr r0 r1 x2 x3 in sy	cs us sy id	
2 0 0	2162	16k	0 0	0 0	0 0 0 0 1 3 0 0	281 461	24 15 40 45
2 0 0	2414	16k	0 0	0 0	0 0 0 0 0 0 0 0	112 646	11 24 76 0
2 0 0	2414	16k	0 0	0 0	0 0 0 0 0 0 0 0	82 704	10 21 79 0
2 0 0	2414	16k	0 0	0 0	0 0 0 0 0 0 0 0	62 754	9 24 76 0
2 0 0	2414	16k	0 0	0 0	0 0 0 0 0 0 0 0	42 791	8 27 73 0
2 0 0	2414	16k	0 0	0 0	0 0 0 0 0 0 0 0	32 823	7 29 71 0
2 0 0	2414	15k	0 0	0 0	0 0 0 0 0 0 0 0	22 846	6 34 66 0
2 0 0	2414	15k	0 0	0 0	0 0 0 0 0 0 0 0	12 867	6 15 85 0
2 0 0	2414	15k	0 0	0 0	0 0 0 0 0 0 0 0	12 881	6 31 69 0
2 0 0	2414	15k	0 0	0 0	0 0 0 0 0 0 0 0	02 896	6 25 75 0

D.4.2.2. VMSTAT -T

243850 reclaims, 1041305344 total time (usec)

average: 4270 usec / reclaim

21572 page ins, 104130534 total time (msec)

average: 4827.1 msec / page in

D.4.2.3. VMSTAT -S

0	swap ins
0	swap outs
41	pages swapped in
107	pages swapped out
2961483	total address trans. faults taken
21572	page ins
0	page outs
177772	pages paged in
0	pages paged out
0	sequential process pages freed
243850	total reclaims (33% fast)
0	reclaims from free list

15	intransit blocking page faults
4663132	zero fill pages created
1836097	zero fill page faults
319180	executable fill pages created
20832	executable fill page faults
2011	swap text pages found in free list
160553	inode text pages found in free list
0	file fill pages created
0	file fill page faults
0	pages examined by the clock daemon
0	revolutions of the clock hand
0	pages freed by the clock daemon
29330671	cpu context switches
155768766	device interrupts
0	pseudo-dma dz interrupts
5503395	traps
1767751554	system calls

D.5. COMMANDE W

D.5.1. MANUEL

NAME

w - display who is logged in and what they are doing

SYNTAX

w [-h] [-s] [-l] [u s e r]

DESCRIPTION

The w command prints a summary of the current activity on the system, including what each user is doing. The heading line shows the current time of day, how long the system has been up, the number of users logged into the system, and the load averages. The load average numbers give the number of jobs in the run queue averaged over 1, 5 and 15 minutes.

The fields output are:

The users login name

The name of the tty the user is on

The time of day the user logged on

The number of minutes since the user last typed anything

The CPU time used by all processes and their children on that terminal

The CPU time used by the currently active processes

The name and arguments of the current process

OPTIONS

-d Outputs debug information.

-h Suppresses the normal header from the output.

-l Displays information in long format (default).

-s Displays information in short format. In the short form, the tty is abbreviated, the login time and cpu times are left off, as are the arguments to commands.

-u Outputs the same information as the `uptime` command.

If a `username` is included, the output will be restricted to that user.

RESTRICTIONS

The notion of the "current process" is unclear. The current algorithm is "the highest numbered process on the terminal that is not ignoring interrupts, or, if there is none, the highest numbered process on the terminal". This fails, for example, in critical sections of programs like the shell and editor, or when faulty programs running in the background fork and fail to ignore interrupts. (In cases where no process can be found, `w` prints "-".)

The CPU time is only an estimate, in particular, if someone leaves a background process running after logging out, the person currently on that terminal is "charged" with the time.

Background processes are not shown, even though they account for much of the load on the system.

Sometimes processes, typically those in the background, are printed with null or garbaged arguments. In these cases, the name of the command is printed in parentheses.

The `w` command does not know about the new conventions for detection of background jobs. It will sometimes find a background job instead of the right one.

FILES

/etc/utmp
/dev/kmem
/dev/drum

SEE ALSO

finger(1), ps(1), who(1)

D.5.2. RESULTAT

```
2:41pm up 14 days, 1 users, load average: 2.42, 2.34, 2.39
User      tty      login@    idle    JCPU  PCPU  what
dedo      tty00    2:38pm   4       w
```

ANNEXE E

LA COMMANDE ETHERFIND SUR SUN

ETHERFIND(8C)

MAINTENANCE COMMANDS

ETHERFIND(8C)

NAME

etherfind - find packets on Ethernet

SYNOPSIS

etherfind [-nprtuvx] [-c -c-o-u-n-t] [-i -i-n-t-e-r-f-a-c-e] -e-x-p-r-e-s-s-i-o-n

AVAILABILITY

This program is available with the -N-e-t-w-o-r-k-i-n-g -T-o-o-l-s -a-n-d -P-r-o-g-r-a-m-s software installation option. Refer to -l-n-s-t-a-l-l-i-n-g -t-h-e-S-u-n-O-S for information on how to install optional software.

DESCRIPTION

etherfind prints out the headers of packets on the ethernet that match the boolean -e-x-p-r-e-s-s-i-o-n. When an internet packet is fragmented into more than one ethernet packet, all fragments except the first are marked with an asterisk. You must be root to invoke etherfind.

OPTIONS

- n Do not convert host addresses and port numbers to names.
- p Normally, the selected interface is put into promiscuous mode, so that etherfind has access to all packets on the ethernet. However, when the -p flag is used, the interface will not go promiscuous.
- r RPC mode: treat each packet as an RPC message, printing the program and procedure numbers.
- t Timestamps: precede each packet listing with a time value in seconds and hundredths of seconds since the first packet.

- u Make the output line buffered.
- v Verbose mode: print out some of the fields of TCP and UDP packets.
- x Dump the header in hex, in addition to the line printed for each packet by default.
- c -c-o-u-n-t
Exit after receiving -c-o-u-n-t packets. This is sometimes useful for dumping a sample of ethernet traffic to a file for later analysis.
- i -i-n-t-e-r-f-a-c-e
etherfind listens on -i-n-t-e-r-f-a-c-e. The program netstat(8C) when invoked with the -i flag lists all the interfaces that a machine has.
- e-x-p-r-e-s-s-i-o-n
The syntax of -e-x-p-r-e-s-s-i-o-n is similar to that used by find(1). Here are the allowable primaries.
- dst -d-e-s-t-i-n-a-t-i-o-n
True if the destination field of the packet is -d-e-s-t-i-n-a-t-i-o-n, which may be either an address or a name.
- src -s-o-u-r-c-e
True if the source field of the packet is -s-o-u-r-c-e, which may be either an address or a name.
- between -h-o-s-t-1 -h-o-s-t-2
True if either the source of the packet is -h-o-s-t-1 and the destination -h-o-s-t-2, or the source is -h-o-s-t-2 and the destination -h-o-s-t-1.
- dstnet -d-e-s-t-i-n-a-t-i-o-n
True if the destination field of the packet has a

network part of `-d-e-s-t-i-n-a-t-i-o-n`, which may be either an address or a name.

`-srcnet -s-o-u-r-c-e`

True if the source field of the packet has a network part of `-s-o-u-r-c-e`, which may be either an address or a name.

`-srcport -p-o-r-t`

True if the packet has a source port value of `-p-o-r-t`. It must be either `udp` or `tcp` (see `tcp(4P)`), `udp(4P)`). The `-p-o-r-t` can be a number or a name used in `/etc/services`.

`-dstport -p-o-r-t`

True if the packet has a destination port value of `-p-o-r-t`. The `-p-o-r-t` can be a number or a name.

`-less -l-e-n-g-t-h`

True if the packet has a length less than or equal to `-l-e-n-g-t-h`.

`-greater -l-e-n-g-t-h`

True if the packet has a length greater than or equal to `-l-e-n-g-t-h`.

`-proto -p-r-o-t-o-c-o-l`

True if the packet is an ip packet (see `ip(4P)`) of protocol type `-p-r-o-t-o-c-o-l`. `-P-r-o-t-o-c-o-l` can be a number or one of the names `icmp`, `udp`, `nd`, or `tcp`.

`-byte -b-y-t-e -o-p -v-a-l-u-e`

True if byte number `-b-y-t-e` of the packet is in relation `-o-p` to `-v-a-l-u-e`. Legal values for `-o-p` are `+`, `<`, `>`, `&`, and `|`. Thus `4=6` is true if the fourth byte of the packet has the value 6, and `20&0xf` is true if byte twenty has one of its four low order bits nonzero.

-broadcast

True if the packet is a broadcast packet.

-arp

True if the packet is a arp packet (see arp(4P)).

-rarp

True if the packet is a rarp packet.

-ip True if the packet is an ip packet.

The primaries may be combined using the following operators
(in order of decreasing precedence):

A parenthesized group of primaries and operators
(parentheses are special to the Shell and must be
escaped).

The negation of a primary ('!' is the unary -n-o-t opera-
tor).

Concatenation of primaries (the -a-n-d operation is
implied by the juxtaposition of two primaries).

Alternation of primaries ('-o' is the -o-r operator).

EXAMPLE

To find all packets arriving at or departing from sundown
example% etherfind -src sundown -o -dst sundown
example%

SEE ALSO

find(1), traffic(1C), arp(4P), ip(4P), nit(4P) tcp(4P),
udp(4P), netstat(8C)

BUGS

The syntax is painful.

ANNEXE F CODE SOURCE DES MODULES

F.1. MODULE WSUN

```
echo 'wsun' > /usr/users/inf/dedo/memoire/controle1
date >> /usr/users/inf/dedo/memoire/controle1
umask 2
cd /usr/users/inf/dedo/memoire
HOME='/usr/users/inf/dedo/memoire'
export HOME
SHELL='/bin/csh'
export SHELL
USER='dedo'
export USER
PATH='/etc:/usr/ucb:/bin:/usr/bin:/usr/local:/usr/hosts:./usr/games:/usr/new:/usr/
local:/usr/local/lib/ean:/usr/local/bin:/usr/src/games/rogue:/usr/new/mh:/usr/new/
lib/mh'
export PATH
SHLVL='1'
export SHLVL
MORE='-c'
export MORE
$SHELL << '...the rest of this file is shell input'
ws1 vmstat 1 10 > /usr/users/inf/dedo/memoire/data/wsun.dat
ws1 vmstat -t >> /usr/users/inf/dedo/memoire/data/wsun.dat
ws1 netstat -i >> /usr/users/inf/dedo/memoire/data/wsun.dat
ws1 vmstat -s >> /usr/users/inf/dedo/memoire/data/wsun.dat
ws1 w >> /usr/users/inf/dedo/memoire/data/wsun.dat
```

F.2. MODULE WSDEUX

```
umask 2
cd /usr/users/inf/dedo/memoire
HOME='/usr/users/inf/dedo/memoire'
export HOME
```



```

SHELL='/bin/csh'
export SHELL
USER='dedo'
export USER
PATH='/etc:/usr/ucb:/bin:/usr/bin:/usr/local:/usr/hosts:/usr/games:/usr/new:/usr/
local:/usr/local/lib/ean:/usr/local/bin:/usr/src/games/rogue:/usr/new/mh:/usr/new/
lib/mh'
export PATH
SHLVL='1'
export SHLVL
MORE='-c'
export MORE
$SHELL << '...the rest of this file is shell input'
echo 'wsdeux' > /usr/users/inf/dedo/memoire/controle2
date >> /usr/users/inf/dedo/memoire/controle2
ws2 vmstat 1 10 > /usr/users/inf/dedo/memoire/data/wsdeux.dat
ws2 vmstat -t >> /usr/users/inf/dedo/memoire/data/wsdeux.dat
ws2 netstat -i >> /usr/users/inf/dedo/memoire/data/wsdeux.dat
ws2 vmstat -s >> /usr/users/inf/dedo/memoire/data/wsdeux.dat
ws2 w >> /usr/users/inf/dedo/memoire/data/wsdeux.dat

```

F.3. MODULE WSTROIS

```

umask 2
cd /usr/users/inf/dedo/memoire
HOME='/usr/users/inf/dedo/memoire'
export HOME
SHELL='/bin/csh'
export SHELL
USER='dedo'
export USER
PATH='/etc:/usr/ucb:/bin:/usr/bin:/usr/local:/usr/hosts:/usr/games:/usr/new:/usr/
local:/usr/local/lib/ean:/usr/local/bin:/usr/src/games/rogue:/usr/new/mh:/usr/new/
lib/mh'
export PATH
SHLVL='1'
export SHLVL
MORE='-c'

```

```

export MORE
$SHELL << '...the rest of this file is shell input'
echo 'wstois' > /usr/users/inf/dedo/memoire/controle3
date >> /usr/users/inf/dedo/memoire/controle3
ws3 vmstat 1 10 > /usr/users/inf/dedo/memoire/data/wstois.dat
ws3 vmstat -t >> /usr/users/inf/dedo/memoire/data/wstois.dat
ws3 netstat -i >> /usr/users/inf/dedo/memoire/data/wstois.dat
ws3 vmstat -s >> /usr/users/inf/dedo/memoire/data/wstois.dat
ws3 w >> /usr/users/inf/dedo/memoire/data/wstois.dat

```

F.4. MODULE WSQUATRE

```

umask 2
cd /usr/users/inf/dedo/memoire
HOME='/usr/users/inf/dedo/memoire'
export HOME
SHELL='/bin/csh'
export SHELL
USER='dedo'
export USER
PATH='/etc:/usr/ucb:/bin:/usr/bin:/usr/local:/usr/hosts:/usr/games:/usr/new:/usr/
local:/usr/local/lib/ean:/usr/local/bin:/usr/src/games/rogue:/usr/new/mh:/usr/new/
lib/mh'
export PATH
SHLVL='1'
export SHLVL
MORE='-c'
export MORE
$SHELL << '...the rest of this file is shell input'
echo 'wsquatre' > /usr/users/inf/dedo/memoire/controle4
date >> /usr/users/inf/dedo/memoire/controle4
ws4 vmstat 1 10 > /usr/users/inf/dedo/memoire/data/wsquatre.dat
ws4 vmstat -t >> /usr/users/inf/dedo/memoire/data/wsquatre.dat
ws4 netstat -i >> /usr/users/inf/dedo/memoire/data/wsquatre.dat
ws4 vmstat -s >> /usr/users/inf/dedo/memoire/data/wsquatre.dat
ws4 w >> /usr/users/inf/dedo/memoire/data/wsquatre.dat

```

F.5. MODULE ELECTRE

```
umask 2
cd /usr/users/inf/dedo/memoire
HOME='/usr/users/inf/dedo/memoire'
export HOME
SHELL='/bin/csh'
export SHELL
USER='dedo'
export USER
PATH='/etc:/usr/ucb:/bin:/usr/bin:/usr/local:/usr/hosts../usr/games:/usr/new:/usr/
local:/usr/local/lib/ean:/usr/local/bin:/usr/src/games/rogue:/usr/new/mh:/usr/new/
lib/mh'
export PATH
SHLVL='1'
export SHLVL
MORE='-c'
export MORE
$SHELL << '...the rest of this file is shell input'
echo 'electre' > /usr/users/inf/dedo/memoire/controlee
date >> /usr/users/inf/dedo/memoire/controlee
electre vmstat 1 10 > /usr/users/inf/dedo/memoire/data/electre.dat
electre vmstat -t >> /usr/users/inf/dedo/memoire/data/electre.dat
electre iostat 1 10 >> /usr/users/inf/dedo/memoire/data/electre.dat
electre df >> /usr/users/inf/dedo/memoire/data/electre.dat
electre netstat -i >> /usr/users/inf/dedo/memoire/data/electre.dat
electre vmstat -s >> /usr/users/inf/dedo/memoire/data/electre.dat
electre w >> /usr/users/inf/dedo/memoire/data/electre.dat
```

F.6. MODULE COMPWS1

```
echo 'compws1' >> /usr/users/inf/dedo/memoire/controle1
date >> /usr/users/inf/dedo/memoire/controle1
cat /usr/users/inf/dedo/memoire/data/init1.dat
/usr/users/inf/dedo/memoire/data/wsun.dat >
/usr/users/inf/dedo/memoire/data/auxun.dat
awk '
    BEGIN{          r=0
```

```

b=0
w=0
us=0
sy=0
id=0
avm=0
free=0
n=0
l1=l2=l3=0
rem=pim=0
inp=iner=out=outer=coll=0 }
{if (NR==1)
    {inp=$1
    inner=$2
    iout=$3
    iouter=$4
    icoll=$5
    ipi=$6
    ipo=$7
    ire=$8
    ifr=$9
    ics=$10
    iint=$11
    ias=$12}
if ((NR>3) && (NR<14))
    {aux=$4
    l=length(aux)
    x=substr(aux,l,1)
    if (x=="k") {aux2=substr(aux,1,l-1)
        aux=aux2*1000}
    if (x=="m") {aux2=substr(aux,1,l-1)
        aux=aux2*1000000}
    r=r+$1
    b=b+$2
    w=w+$3
    us=us+$20
    sy=sy+$21
    id=id+$22

```

```

    avm=avm+aux
    aux=$5
    l=length(aux)
    x=substr(aux,l,1)
    if (x=="k") {aux2=substr(aux,1,l-1)
        aux=aux2*1000}
    if (x=="m") {aux2=substr(aux,1,l-1)
        aux=aux2*1000000}
    free=free+aux}
if (NR==15)
    {rem=$2}
if (NR==18)
    {pim=$2}
if (NR==20)
    if ($5>=iinp) {inp=$5-iinp}      else {inp=$5}
    if ($6>=iiner) {iner=$6-iiner}  else {iner=$6}
    if ($7>=iout) {out=$7-iout}     else {out=$7}
    if ($8>=iouter) {outer=$8-iouter} else {outer=$8}
    if ($9>=icoll) {coll=$9-icoll}  else {coll=$9}}
if (NR==30) {if ($1>=ipi) {pi=$1-ipi} else {pi=$1}}
if (NR==31) {if ($1>=ipo) {po=$1-ipo} else {po=$1}}
if (NR==34) {if ($1>=ire) {re=$1-ire} else {re=$1}}
if (NR==46) {if ($1>=ifr) {fr=$1-ifr} else {fr=$1}}
if (NR==47) {if ($1>=ics) {cs=$1-ics} else {cs=$1}}
if (NR==48) {if ($1>=iint) {zint=$1-iint} else {zint=$1}}
if (NR==51) {if ($1>=ias) {as=$1-ias} else {as=$1}}
if (NR==52)
    {l1=$(NF-2)
    l2=$(NF-1)
    l3=$NF}}
END{
    r=r/10
    b=b/10
    w=w/10
    us=us/10
    sy=sy/10
    id=id/10
    avm=avm/10

```



```

n=0
l1=l2=l3=0
rem=pim=0
inp=iner=out=outer=coll=0 }
{if (NR==1)
  {inp=$1
  iiner=$2
  iout=$3
  iouter=$4
  icoll=$5
  ipi=$6
  ipo=$7
  ire=$8
  ifr=$9
  ics=$10
  iint=$11
  ias=$12}
  if ((NR>3) && (NR<14))
    {aux=$4
    l=length(aux)
    x=substr(aux,l,1)
    if (x=="k") {aux2=substr(aux,1,l-1)
      aux=aux2*1000}
    if (x=="m") {aux2=substr(aux,1,l-1)
      aux=aux2*1000000}
    r=r+$1
    b=b+$2
    w=w+$3
    us=us+$20
    sy=sy+$21
    id=id+$22
    avm=avm+aux
    aux=$5
    l=length(aux)
    x=substr(aux,l,1)
    if (x=="k") {aux2=substr(aux,1,l-1)
      aux=aux2*1000}
    if (x=="m") {aux2=substr(aux,1,l-1)

```



```

        printf("%s\t%s\t%s\t%s\t%s\n",inp,iner,out,outer,coll)
}' /usr/users/inf/dedo/memoire/data/auxdeux.dat >>
/usr/users/inf/dedo/memoire/data/compws2.dat
awk '
    {if (NR==19) {printf("%s\t%s\t%s\t%s\t%s\t", $5,$6,$7,$8,$9)}
    if (NR==29) {printf("%s\t", $1)}
    if (NR==30) {printf("%s\t", $1)}
    if (NR==33) {printf("%s\t", $1)}
    if (NR==45) {printf("%s\t", $1)}
    if (NR==46) {printf("%s\t", $1)}
    if (NR==47) {printf("%s\t", $1)}
    if (NR==50) {printf("%s\n", $1)}
}' /usr/users/inf/dedo/memoire/data/wsdeux.dat >
/usr/users/inf/dedo/memoire/data/init2.dat

```

F.8. MODULE COMPWS3

```

echo 'compws3' >> /usr/users/inf/dedo/memoire/controle3
date >> /usr/users/inf/dedo/memoire/controle3
cat /usr/users/inf/dedo/memoire/data/init3.dat
/usr/users/inf/dedo/memoire/data/wsttrois.dat >
/usr/users/inf/dedo/memoire/data/auxtrois.dat
awk '

```

```

    BEGIN{          r=0
        b=0
        w=0
        us=0
        sy=0
        id=0
        avm=0
        free=0
        n=0
        l1=l2=l3=0
        rem=pim=0
        inp=iner=out=outer=coll=0 }
    {if (NR==1)
        {inp=$1
        iiner=$2

```

```

iout=$3
iouter=$4
icoll=$5
ipi=$6
ipo=$7
ire=$8
ifr=$9
ics=$10
iint=$11
ias=$12}
if ((NR>3) && (NR<14))
    {aux=$4
    l=length(aux)
    x=substr(aux,l,1)
    if (x=="k") {aux2=substr(aux,1,l-1)
        aux=aux2*1000}
    if (x=="m") {aux2=substr(aux,1,l-1)
        aux=aux2*1000000}
    r=r+$1
    b=b+$2
    w=w+$3
    us=us+$20
    sy=sy+$21
    id=id+$22
    avm=avm+aux
    aux=$5
    l=length(aux)
    x=substr(aux,l,1)
    if (x=="k") {aux2=substr(aux,1,l-1)
        aux=aux2*1000}
    if (x=="m") {aux2=substr(aux,1,l-1)
        aux=aux2*1000000}
    free=free+aux}
if (NR==15)
    {rem=$2}
if (NR==18)
    {pim=$2}
if (NR==20)

```

```

        if ($5>=iinp) {inp=$5-iinp}           else {inp=$5}
        if ($6>=iiner) {iner=$6-iiner}       else {iner=$6}
        if ($7>=iout) {out=$7-iout}          else {out=$7}
        if ($8>=iouter) {outer=$8-iouter}    else {outer=$8}
        if ($9>=icoll) {coll=$9-icoll}       else {coll=$9}

if (NR==30) {if ($1>=ipi) {pi=$1-ipi}      else {pi=$1}}
if (NR==31) {if ($1>=ipo) {po=$1-ipo}     else {po=$1}}
if (NR==34) {if ($1>=ire) {re=$1-ire}     else {re=$1}}
if (NR==46) {if ($1>=ifr) {fr=$1-ifr}     else {fr=$1}}
if (NR==47) {if ($1>=ics) {cs=$1-ics}     else {cs=$1}}
if (NR==48) {if ($1>=iint) {zint=$1-iint} else {zint=$1}}
if (NR==51) {if ($1>=ias) {as=$1-ias}     else {as=$1}}
if (NR==52)
    {l1=$(NF-2)
    l2=$(NF-1)
    l3=$NF}}
END{
    r=r/10
    b=b/10
    w=w/10
    us=us/10
    sy=sy/10
    id=id/10
    avm=avm/10
    free=free/10
    l1=substr(l1,1,4)
    l2=substr(l2,1,4)
    printf("%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\n",
           r,b,w,us,sy,id,zint,as,cs,l1,l2,l3)
    printf("%s\t%s\n",avm,free)
    printf("%s\t%s\t%s\t%s\t%s\t%s\n",re,pi,po,fr,rem,pim)
    printf("%s\t%s\t%s\t%s\t%s\n",inp,iner,out,outer,coll)
}' /usr/users/inf/dedo/memoire/data/auxtrois.dat >>
   /usr/users/inf/dedo/memoire/data/compws3.dat
awk '
if (NR==19) {printf("%s\t%s\t%s\t%s\t%s\t", $5,$6,$7,$8,$9)}
if (NR==29) {printf("%s\t", $1)}
if (NR==30) {printf("%s\t", $1)}

```

```

if (NR==33) {printf("%s\t",$1)}
if (NR==45) {printf("%s\t",$1)}
if (NR==46) {printf("%s\t",$1)}
if (NR==47) {printf("%s\t",$1)}
if (NR==50) {printf("%s\n",$1)}
}' /usr/users/inf/dedo/memoire/data/wstois.dat >
/usr/users/inf/dedo/memoire/data/init3.dat

```

F.9. MODULE COMPWS4

```

echo 'compws4' >> /usr/users/inf/dedo/memoire/controle4
date >> /usr/users/inf/dedo/memoire/controle4
cat /usr/users/inf/dedo/memoire/data/init4.dat
/usr/users/inf/dedo/memoire/data/wsquatre.dat >
/usr/users/inf/dedo/memoire/data/auxquatre.dat
awk '

```

```

    BEGIN{          r=0
        b=0
        w=0
        us=0
        sy=0
        id=0
        avm=0
        free=0
        n=0
        l1=l2=l3=0
        rem=pim=0
        inp=iner=out=outer=coll=0 }
    {if (NR==1)
        {iinp=$1
        iiner=$2
        iout=$3
        iouter=$4
        icoll=$5
        ipi=$6
        ipo=$7
        ire=$8
        ifr=$9

```

```

ics=$10
iint=$11
ias=$12}
if ((NR>3) && (NR<14))
{aux=$4
l=length(aux)
x=substr(aux,l,1)
if (x=="k") {aux2=substr(aux,1,l-1)
aux=aux2*1000}
if (x=="m") {aux2=substr(aux,1,l-1)
aux=aux2*1000000}
r=r+$1
b=b+$2
w=w+$3
us=us+$20
sy=sy+$21
id=id+$22
avm=avm+aux
aux=$5
l=length(aux)
x=substr(aux,l,1)
if (x=="k") {aux2=substr(aux,1,l-1)
aux=aux2*1000}
if (x=="m") {aux2=substr(aux,1,l-1)
aux=aux2*1000000}
free=free+aux}
if (NR==15)
{rem=$2}
if (NR==18)
{pim=$2}
if (NR==20)
{if ($5>=iinp) {inp=$5-iinp} else {inp=$5}
if ($6>=iiner) {iner=$6-iiner} else {iner=$6}
if ($7>=iout) {out=$7-iout} else {out=$7}
if ($8>=iouter) {outer=$8-iouter} else {outer=$8}
if ($9>=icoll) {coll=$9-icoll} else {coll=$9}}
if (NR==30) {if ($1>=ipi) {pi=$1-ipi} else {pi=$1}}
if (NR==31) {if ($1>=ipo) {po=$1-ipo} else {po=$1}}

```

```

if (NR==34) {if ($1>=ire) {re=$1-ire}    else {re=$1}}
if (NR==46) {if ($1>=ifr) {fr=$1-ifr}    else {fr=$1}}
if (NR==47) {if ($1>=ics) {cs=$1-ics}    else {cs=$1}}
if (NR==48) {if ($1>=iint) {zint=$1-iint} else {zint=$1}}
if (NR==51) {if ($1>=ias) {as=$1-ias}    else {as=$1}}
if (NR==52)
    {l1=$(NF-2)
    l2=$(NF-1)
    l3=$NF}}
END{
    r=r/10
    b=b/10
    w=w/10
    us=us/10
    sy=sy/10
    id=id/10
    avm=avm/10
    free=free/10
    l1=substr(l1,1,4)
    l2=substr(l2,1,4)
    printf("%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\n",
           r,b,w,us,sy,id,zint,as,cs,l1,l2,l3)
    printf("%s\t%s\n",avm,free)
    printf("%s\t%s\t%s\t%s\t%s\t%s\t%s\n",re,pi,po,fr,rem,pim)
    printf("%s\t%s\t%s\t%s\t%s\t%s\n",inp,iner,out,outer,coll)
}' /usr/users/inf/dedo/memoire/data/auxquatre.dat >>
/usr/users/inf/dedo/memoire/data/compws4.dat
awk '
{if (NR==19) {printf("%s\t%s\t%s\t%s\t%s\t", $5,$6,$7,$8,$9)}
if (NR==29) {printf("%s\t", $1)}}
if (NR==30) {printf("%s\t", $1)}}
if (NR==33) {printf("%s\t", $1)}}
if (NR==45) {printf("%s\t", $1)}}
if (NR==46) {printf("%s\t", $1)}}
if (NR==47) {printf("%s\t", $1)}}
if (NR==50) {printf("%s\n", $1)}}
}' /usr/users/inf/dedo/memoire/data/wsquatre.dat >
/usr/users/inf/dedo/memoire/data/init4.dat

```

F.10. MODULE COMPELECTRE

```
echo 'compelectre' >> /usr/users/inf/dedo/memoire/controlee
date >> /usr/users/inf/dedo/memoire/controlee
cat /usr/users/inf/dedo/memoire/data/inite.dat
/usr/users/inf/dedo/memoire/data/electre.dat >
/usr/users/inf/dedo/memoire/data/auxe.dat
awk '
```

```
    BEGIN{          r=0
                b=0
                w=0
                us=0
                sy=0
                id=0
                avm=0
                free=0
                n=0
                l1=l2=l3=0
                inp=iner=out=outer=coll=0
                rem=pim=0 }
    {if (NR==1)
        {iinp=$1
          iiner=$2
          iout=$3
          iouter=$4
          icoll=$5
          ipi=$6
          ipo=$7
          ire=$8
          ifr=$9
          ics=$10
          iint=$11
          ias=$12}
    if ((NR>3) && (NR<14))
        {aux=$4
          l=length(aux)
          x=substr(aux,l,1)
          if (x=="k") {aux2=substr(aux,1,l-1)
                        aux=aux2*1000}
```

```

if (x=="m") {aux2=substr(aux,1,l-1)
    aux=aux2*1000000}
r=r+$1
b=b+$2
w=w+$3
us=us+$20
sy=sy+$21
id=id+$22
avm=avm+aux
aux=$5
l=length(aux)
x=substr(aux,l,1)
if (x=="k") {aux2=substr(aux,1,l-1)
    aux=aux2*1000}
if (x=="m") {aux2=substr(aux,1,l-1)
    aux=aux2*1000000}
free=free+aux
ops0=ops0+$13
ops1=ops1+$14}
if (NR==15) {rem=$2}
if (NR==18) {pim=$2}
if ((NR>20) && (NR<31))
    {bps0=bps0+$3
    tps0=tps0+$4
    bps1=bps1+$5
    tps1=tps1+$6}
if (NR==33) {base=$5}
if (NR==34) {usr=$5}
if (NR==35) {var=$5}
if (NR==36) {tmp=$5}
if (NR==37) {users=$5}
if (NR==40) {inf=$5}
if (NR==43)
    {if ($5>=iinp) {inp=$5-iinp}      else {inp=$5}
    if ($6>=iiner) {iner=$6-iiner}    else {iner=$6}
    if ($7>=iout) {out=$7-iout}       else {out=$7}
    if ($8>=iouter) {outer=$8-iouter} else {outer=$8}
    if ($9>=icoll) {coll=$9-icoll}    else {coll=$9}}

```



```

if (NR==52) {if ($1>=ipi) {pi=$1-ipi}      else {pi=$1}}
if (NR==53) {if ($1>=ipo) {po=$1-ipo}     else {po=$1}}
if (NR==56) {if ($1>=ire) {re=$1-ire}     else {re=$1}}
if (NR==68) {if ($1>=ifr) {fr=$1-ifr}     else {fr=$1}}
if (NR==69) {if ($1>=ics) {cs=$1-ics}     else {cs=$1}}
if (NR==70) {if ($1>=iint) {zint=$1-iint} else {zint=$1}}
if (NR==73) {if ($1>=ias) {as=$1-ias}     else {as=$1}}
if (NR==74) { l1=$(NF-2)
               l2=$(NF-1)
               l3=$NF }
}
END{
    r=r/10
    b=b/10
    w=w/10
    us=us/10
    sy=sy/10
    id=id/10
    avm=avm/10
    free=free/10
    l1=substr(l1,1,4)
    l2=substr(l2,1,4)
    ops0=ops0/10
    ops1=ops1/10
    bps0=bps0/10
    tps0=tps0/10
    bps1=bps1/10
    tps1=tps1/10
    L=length(base)
    base=substr(base,1,L-1)
    L=length(usr)
    usr=substr(usr,1,L-1)
    L=length(var)
    var=substr(var,1,L-1)
    L=length(tmp)
    tmp=substr(tmp,1,L-1)
    L=length(users)
    users=substr(users,1,L-1)
}

```

```

L=length(inf)
inf=substr(inf,1,L-1)
printf("%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\n",
        r,b,w,us,sy,id,zint,as,cs,l1,l2,l3)
printf("%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\n",
        ops0,bps0,tps0,ops1,bps1,tps1,base,usr,var,tmp,users,inf)
printf("%s\t%s\n",avm,free)
printf("%s\t%s\t%s\t%s\t%s\t%s\n",re,pi,po,fr,rem,pim)
printf("%s\t%s\t%s\t%s\t%s\n",inp,iner,out,outer,coll)
}' /usr/users/inf/dedo/memoire/data/auxe.dat >>
/usr/users/inf/dedo/memoire/data/compelectre.dat
awk '
{if (NR==42) {printf("%s\t%s\t%s\t%s\t%s\t", $5,$6,$7,$8,$9)}
if (NR==51) {printf ("%s\t", $1)}}
if (NR==52) {printf ("%s\t", $1)}}
if (NR==55) {printf ("%s\t", $1)}}
if (NR==67) {printf ("%s\t", $1)}}
if (NR==68) {printf ("%s\t", $1)}}
if (NR==69) {printf ("%s\t", $1)}}
if (NR==72) {printf ("%s\n", $1)}}
}' /usr/users/inf/dedo/memoire/data/electre.dat >
/usr/users/inf/dedo/memoire/data/inite.dat

```

F.11. MODULE COMPHEURE

```

echo 'compheure' >> /usr/users/inf/dedo/memoire/controle
date >> /usr/users/inf/dedo/memoire/controle
cat /usr/users/inf/dedo/memoire/data/compelectre.dat
/usr/users/inf/dedo/memoire/data/compws1.dat
/usr/users/inf/dedo/memoire/data/compws2.dat
/usr/users/inf/dedo/memoire/data/compws3.dat
/usr/users/inf/dedo/memoire/data/compws4.dat >
/usr/users/inf/dedo/memoire/data/aux.dat ;
awk '
    BEGIN
        {re=r1=r2=r3=r4=0
        be=b1=b2=b3=b3=0
        we=w1=w2=w3=w4=0

```

```

use=us1=us2=us3=us4=0
sy=sy1=sy2=sy3=sy4=0
ide=id1=id2=id3=id4=0
ie=i1=i2=i3=i4=0
ae=a1=a2=a3=a4=0
cse=cs1=cs2=cs3=cs4=0
l1e=l11=l12=l13=l14=0
l2e=l21=l22=l23=l24=0
l3e=l31=l32=l33=l34=0
ops0=bps0=tps0=ops1=bps1=tps1=0
base=usr=var=tmp=users=students=0
nume=num1=num2=num3=num4=test=0
i=1}
{if ((NR==1)|| (NR==6)|| (NR==11)|| (NR==16))
    {if (($4==0)&&($5==0)&&($6==0)) {nume=nume+1}
    else {
        re=re+$1
        be=be+$2
        we=we+$3
        use=use+$4
        sye=sye+$5
        ide=ide+$6
        ie=ie+$7
        ae=ae+$8
        cse=cse+$9
        l1e=l1e+$10
        l2e=l2e+$11
        l3e=l3e+$12}}
if ((NR==2)|| (NR==7)|| (NR==12)|| (NR==17))
    {ops0=ops0+$1
    bps0=bps0+$2
    tps0=tps0+$3
    ops1=ops1+$4
    bps1=bps1+$5
    tps1=tps1+$6
    base=base+$7
    usr=usr+$8
    var=var+$9

```

```

tmp=tmp+$10
users=users+$11
students=students+$12}
if ((NR==21)||(NR==25)||(NR==29)||(NR==33))
{if (($4==0)&&($5==0)&&($6==0)) {num1=num1+1}
else {r1=r1+$1
      b1=b1+$2
      w1=w1+$3
      us1=us1+$4
      sy1=sy1+$5
      id1=id1+$6
      i1=i1+$7
      a1=a1+$8
      cs1=cs1+$9
      l11=l11+$10
      l21=l21+$11
      l31=l31+$12}}
if ((NR==37)||(NR==41)||(NR==45)||(NR==49))
{if (($4==0)&&($5==0)&&($6==0)) {num2=num2+1}
else {r2=r2+$1
      b2=b2+$2
      w2=w2+$3
      us2=us2+$4
      sy2=sy2+$5
      id2=id2+$6
      i2=i2+$7
      a2=a2+$8
      cs2=cs2+$9
      l12=l12+$10
      l22=l22+$11
      l32=l32+$12}}
if ((NR==53)||(NR==57)||(NR==61)||(NR==65))
{if (($4==0)&&($5==0)&&($6==0)) {num3=num3+1}
else {r3=r3+$1
      b3=b3+$2
      w3=w3+$3
      us3=us3+$4
      sy3=sy3+$5

```

```

        id3=id3+$6
        i3=i3+$7
        a3=a3+$8
        cs3=cs3+$9
        l13=l13+$10
        l23=l23+$11
        l33=l33+$12}}
if ((NR==69)||((NR==73)||((NR==77)||((NR==81))
    {if (($4==0)&&($5==0)&&($6==0)) {num4=num4+1}
    else {r4=r4+$1
        b4=b4+$2
        w4=w4+$3
        us4=us4+$4
        sy4=sy4+$5
        id4=id4+$6
        i4=i4+$7
        a4=a4+$8
        cs4=cs4+$9
        l14=l14+$10
        l24=l24+$11
        l34=l34+$12}}
}
END {nume=int(num4/4)
    dive=4-nume
    num1=int(num1/4)
    div1=4-num1
    num2=int(num2/4)
    div2=4-num2
    num3=int(num3/4)
    div3=4-num3
    num4=int(num4/4)
    div4=4-num4
    re=re/dive
    be=be/dive
    we=we/dive
    use=use/dive
    sye=sye/dive
    ide=ide/dive

```

$l1e=l1e/div1$
 $l2e=l2e/div1$
 $l3e=l3e/div1$
 $r1=r1/div1$
 $b1=b1/div1$
 $w1=w1/div1$
 $us1=us1/div1$
 $sy1=sy1/div1$
 $id1=id1/div1$
 $l11=l11/div1$
 $l21=l21/div1$
 $l31=l31/div1$
 $r2=r2/div2$
 $b2=b2/div2$
 $w2=w2/div2$
 $us2=us2/div2$
 $sy2=sy2/div2$
 $id2=id2/div2$
 $l12=l12/div2$
 $l22=l22/div2$
 $l32=l32/div2$
 $r3=r3/div3$
 $b3=b3/div3$
 $w3=w3/div3$
 $us3=us3/div3$
 $sy3=sy3/div3$
 $id3=id3/div3$
 $l13=l13/div3$
 $l23=l23/div3$
 $l33=l33/div3$
 $r4=r4/div4$
 $b4=b4/div4$
 $w4=w4/div4$
 $us4=us4/div4$
 $sy4=sy4/div4$
 $id4=id4/div4$
 $l14=l14/div4$
 $l24=l24/div4$

```

l34=l34/div4
ops0=ops0/dive
bps0=bps0/dive
tps0=tps0/dive
ops1=ops1/dive
bps1=bps1/dive
tps1=tps1/dive
base=base/dive
usr=usr/dive
var=var/dive
tmp=tmp/dive
users=users/dive
students=students/dive

```

```

printf("CPU\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t",
       re,be,we,use,sye,ide,ie,ae,cse,l1e,l2e,l3e)
printf("\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t",
       r1,b1,w1,us1,sy1,id1,i1,a1,cs1,l11,l21,l31)
printf("\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t",
       r2,b2,w2,us2,sy2,id2,i2,a2,cs2,l12,l22,l32)
printf("\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t",
       r3,b3,w3,us3,sy3,id3,i3,a3,cs3,l13,l23,l33)
printf("\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t",
       r4,b4,w4,us4,sy4,id4,i4,a4,cs4,l14,l24,l34)
printf("DISQUE\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t",
       ops0,bps0,tps0,ops1,bps1,tps1,base,usr,var,tmp,users,students)
} ' /usr/users/inf/dedo/memoire/data/aux.dat >
    /usr/users/inf/dedo/memoire/data/heure.dat

```

awk '

```

BEGIN{ avme=avm1=avm2=avm3=avm4=0
       freee=free1=free2=free3=free4=0
       ree=re1=re2=re3=re4=0
       pie=pi1=pi2=pi3=pi4=0
       poe=po1=po2=po3=po4=0
       fre=fr1=fr2=fr3=fr4=0
       reme=rem1=rem2=rem3=rem4=0
       pime=pim1=pim2=pim3=pim4=0
       inpe=inp1=inp2=inp3=inp4=0

```

```

inere=iner1=inere2=iner3=iner4=0
oute=out1=out2=out3=out4=0
outere=outer1=outer2=outer3=outer4=0
colle=coll1=coll2=coll3=coll4=0
nume=num1=num2=num3=num4=0
}
{if ((NR==3)||(NR==8)||(NR==13)||(NR==18))
    {avme=avme+$1
    freee=freee+$2}
if ((NR==22)||(NR==26)||(NR==30)||(NR==34))
    {avm1=avm1+$1
    free1=free1+$2}
if ((NR==38)||(NR==42)||(NR==46)||(NR==50))
    {avm2=avm2+$1
    free2=free2+$2}
if ((NR==54)||(NR==58)||(NR==62)||(NR==66))
    {avm3=avm3+$1
    free3=free3+$2}
if ((NR==70)||(NR==74)||(NR==78)||(NR==82))
    {avm4=avm4+$1
    free4=free4+$2}
if ((NR==4)||(NR==9)||(NR==14)||(NR==19))
    {ree=ree+$1
    pie=pie+$2
    poe=poe+$3
    fre=fre+$4
    reme=reme+$5
    pime=pime+$6}
if ((NR==23)||(NR==27)||(NR==31)||(NR==35))
    {re1=re1+$1
    pi1=pi1+$2
    po1=po1+$3
    fr1=fr1+$4
    rem1=rem1+$5
    pim1=pim1+$6}
if ((NR==39)||(NR==43)||(NR==47)||(NR==51))
    {re2=re2+$1
    pi2=pi2+$2

```



```

po2=po2+$3
fr2=fr2+$4
rem2=rem2+$5
pim2=pim2+$6}
if ((NR==55)||(NR==59)||(NR==63)||(NR==67))
{re3=re3+$1
pi3=pi3+$2
po3=po3+$3
fr3=fr3+$4
rem3=rem3+$5
pim3=pim3+$6}
if ((NR==71)||(NR==75)||(NR==79)||(NR==83))
{re4=re4+$1
pi4=pi4+$2
po4=po4+$3
fr4=fr4+$4
rem4=rem4+$5
pim4=pim4+$6}
if ((NR==5)||(NR==10)||(NR==15)||(NR==20))
{inpe=inpe+$1
inere=inere+$2
oute=oute+$3
outere=outere+$4
colle=colle=$5}
if ((NR==24)||(NR==28)||(NR==32)||(NR==36))
{inp1=inp1+$1
iner1=iner1+$2
out1=out1+$3
outer1=outer1+$4
coll1=coll1=$5}
if ((NR==40)||(NR==44)||(NR==48)||(NR==52))
{inp2=inp2+$1
iner2=iner2+$2
out2=out2+$3
outer2=outer2+$4
coll2=coll2=$5}
if ((NR==56)||(NR==60)||(NR==64)||(NR==68))
{inp3=inp3+$1

```

```

    iner3=iner3+$2
    out3=out3+$3
    outer3=outer3+$4
    coll3=coll3=$5}
if ((NR==72)|| (NR==76)|| (NR==80)|| (NR==84))
    {inp4=inp4+$1
    iner4=iner4+$2
    out4=out4+$3
    outer4=outer4+$4
    coll4=coll4=$5}
}
END {if (nume!=4) {
        dive=4-nume
        avme=avme/dive
        }
    if (num1!=4) {
        div1=4-num1
        avm1=avm1/div1
        }
    if (num2!=4) {
        div2=4-num2
        avm2=avm2/div2
        }
    if (num3!=4) {
        div3=4-num3
        avm3=avm3/div3
        }
    if (num4!=4) {
        div4=4-num4
        avm4=avm4/div4
        }
    freee=freee/dive
    reme=reme/dive
    pime=pime/dive
    free1=free1/div1
    rem1=rem1/div1
    pim1=pim1/div1
    free2=free2/div2

```

```

rem2=rem2/div2
pim2=pim2/div2
free3=free3/div3
rem3=rem3/div3
pim3=pim3/div3
free4=free4/div4
rem4=rem4/div4
pim4=pim4/div4
printf("MEMOIRE\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\n",
      avme,freee,avm1,free1,avm2,free2,avm3,free3,avm4,free4)
printf("PAGE\t%s\t%s\t%s\t%s\t%s\t%s\t",ree,pie,poe,fre,reme,pime)
printf("%s\t%s\t%s\t%s\t%s\t%s\t",re1,pi1,po1,fr1,rem1,pim1)
printf("%s\t%s\t%s\t%s\t%s\t%s\t",re2,pi2,po2,fr2,rem2,pim2)
printf("%s\t%s\t%s\t%s\t%s\t%s\t",re3,pi3,po3,fr3,rem3,pim3)
printf("%s\t%s\t%s\t%s\t%s\t%s\n",re4,pi4,po4,fr4,rem4,pim4)
printf("RESEAU\t%s\t%s\t%s\t%s\t%s\t",inpe,inere,outere,oute,outer)
printf("%s\t%s\t%s\t%s\t%s\t",inp1,iner1,out1,outer1,coll1)
printf("%s\t%s\t%s\t%s\t%s\t",inp2,iner2,out2,outer2,coll2)
printf("%s\t%s\t%s\t%s\t%s\t",inp3,iner3,out3,outer3,coll3)
printf("%s\t%s\t%s\t%s\t%s\n",inp4,iner4,out4,outer4,coll4)
} ' /usr/users/inf/dedo/memoire/data/aux.dat >>
      /usr/users/inf/dedo/memoire/data/heure.dat

```

```

cat /usr/users/inf/dedo/memoire/data/heure.dat
      /usr/users/inf/dedo/memoire/performance.jc >
      /usr/users/inf/dedo/memoire/data/aux2.dat

```

```

awk '
BEGIN{d=2
      m=2
      p=2
      r=2
      i=0
      n=0}
{if (NR==1) {l1=$0}
  if (NR==2) {l2=$0}
  if (NR==3) {l3=$0}
  if (NR==4) {l4=$0}
  if (NR==5) {l5=$0}
  if ($1=="CPU") {n++}
}

```

```

if (($1=="DISQUE")&&(d==0))
    {d=1
    print l1
    while (i<(24-n))
        {print " "
        i++}
    i=0}
if (($1=="MEMOIRE")&&(m==0))
    {m=1
    print l2
    while (i<(24-n))
        {print " "
        i++}
    i=0}
if (($1=="PAGE")&&(p==0))
    {p=1
    print l3
    while (i<(24-n))
        {print " "
        i++}
    i=0}
if (($1=="RESEAU")&&(r==0))
    {r=1
    print l4
    while (i<(24-n))
        {print " "
        i++}
    i=0}
if ((NR>5)&&(NF>1)) {print $0}
if (NR==5)
    {d=0
    m=0
    p=0
    r=0}
}
END{if (NR<6)
    {print l1
    n=NR-4

```

```

        while (i<(24-n))
            {print " "
             i=i+1}
        i=0
        print l2
        while (i<(24-n))
            {print " "
             i=i+1}
        i=0
        print l3
        while (i<(24-n))
            {print " "
             i=i+1}
        i=0
        print l4
        while (i<(24-n))
            {print " "
             i=i+1}
        i=0}
    print l5
    while (i<(24-n))
        {print " "
         i=i+1}
} ' /usr/users/inf/dedo/memoire/data/aux2.dat >
    /usr/users/inf/dedo/memoire/performance.jc
cp /usr/users/inf/dedo/memoire/data/vide.dat
    /usr/users/inf/dedo/memoire/data/compws1.dat
cp /usr/users/inf/dedo/memoire/data/vide.dat
    /usr/users/inf/dedo/memoire/data/compws2.dat
cp /usr/users/inf/dedo/memoire/data/vide.dat
    /usr/users/inf/dedo/memoire/data/compws3.dat
cp /usr/users/inf/dedo/memoire/data/vide.dat
    /usr/users/inf/dedo/memoire/data/compws4.dat
cp /usr/users/inf/dedo/memoire/data/vide.dat
    /usr/users/inf/dedo/memoire/data/compelectre.dat

```

F.12. MODULE COMPJOUR

```
echo 'compjour' >> /usr/users/inf/dedo/memoire/controle
date >> /usr/users/inf/dedo/memoire/controle
cp /usr/users/inf/dedo/memoire/performance.jc
/usr/users/inf/dedo/memoire/performance.jp
awk '
```

```
    BEGIN{re=r1=r2=r3=r4=0
        be=b1=b2=b3=b3=0
        we=w1=w2=w3=w4=0
        use=us1=us2=us3=us4=0
        sye=sy1=sy2=sy3=sy4=0
        ide=id1=id2=id3=id4=0
        ie=i1=i2=i3=i4=0
        ae=a1=a2=a3=a4=0
        cse=cs1=cs2=cs3=cs4=0
        l1e=l11=l12=l13=l14=0
        l2e=l21=l22=l23=l24=0
        l3e=l31=l32=l33=l34=0
        ops0=bps0=tps0=ops1=bps1=tps1=0
        base=usr=var=tmp=users=students=0
    }
    {if ((NR>=1)&&(NR<=24))
        {re=re+$2
        be=be+$3
        we=we+$4
        use=use+$5
        sye=sye+$6
        ide=ide+$7
        ie=ie+$8
        ae=ae+$9
        cse=cse+$10
        l1e=l1e+$11
        l2e=l2e+$12
        l3e=l3e+$13
        r1=r1+$14
        b1=b1+$15
        w1=w1+$16
        us1=us1+$17
```

sy1=sy1+\$18
id1=id1+\$19
i1=i1+\$20
a1=a1+\$21
cs1=cs1+\$22
l11=l11+\$23
l21=l21+\$24
l31=l31+\$25
r2=r2+\$26
b2=b2+\$27
w2=w2+\$28
us2=us2+\$29
sy2=sy2+\$30
id2=id2+\$31
i2=i2+\$32
a2=a2+\$33
cs2=cs2+\$34
l12=l12+\$35
l22=l22+\$36
l32=l32+\$37
r3=r3+\$38
b3=b3+\$39
w3=w3+\$40
us3=us3+\$41
sy3=sy3+\$42
id3=id3+\$43
i3=i3+\$44
a3=a3+\$45
cs3=cs3+\$46
l13=l13+\$47
l23=l23+\$48
l33=l33+\$49
r4=r4+\$50
b4=b4+\$51
w4=w4+\$52
us4=us4+\$53
sy4=sy4+\$54
id4=id4+\$55

```

i4=i4+$56
a4=a4+$57
cs4=cs4+$58
l14=l14+$59
l24=l24+$60
l34=l34+$61}
if ((NR>=25)&&(NR<=48))
{ops0=ops0+$2
bps0=bps0+$3
tps0=tps0+$4
ops1=ops1+$5
bps1=bps1+$6
tps1=tps1+$7
base=base+$8
usr=usr+$9
var=var+$10
tmp=tmp+$11
users=users+$12
students=students+$13}
}
END {re=re/24
be=be/24
we=we/24
use=use/24
sye=sye/24
ide=ide/24
ie=ie/24
ae=ae/24
cse=cse/24
l1e=l1e/24
l2e=l2e/24
l3e=l3e/24
r1=r1/24
b1=b1/24
w1=w1/24
us1=us1/24
sy1=sy1/24
id1=id1/24

```


$i1=i1/24$
 $a1=a1/24$
 $cs1=cs1/24$
 $l11=l11/24$
 $l21=l21/24$
 $l31=l31/24$
 $r2=r2/24$
 $b2=b2/24$
 $w2=w2/24$
 $us2=us2/24$
 $sy2=sy2/24$
 $id2=id2/24$
 $i2=i2/24$
 $a2=a2/24$
 $cs2=cs2/24$
 $l12=l12/24$
 $l22=l22/24$
 $l32=l32/24$
 $r3=r3/24$
 $b3=b3/24$
 $w3=w3/24$
 $us3=us3/24$
 $sy3=sy3/24$
 $id3=id3/24$
 $i3=i3/24$
 $a3=a3/24$
 $cs3=cs3/24$
 $l13=l13/24$
 $l23=l23/24$
 $l33=l33/24$
 $r4=r4/24$
 $b4=b4/24$
 $w4=w4/24$
 $us4=us4/24$
 $sy4=sy4/24$
 $id4=id4/24$
 $i4=i4/24$
 $a4=a4/24$

```

cs4=cs4/24
l14=l14/24
l24=l24/24
l34=l34/24
ops0=ops0/24
bps0=bps0/24
tps0=tps0/24
ops1=ops1/24
bps1=bps1/24
tps1=tps1/24
base=base/24
usr=usr/24
var=var/24
tmp=tmp/24
users=users/24
students=students/24
printf("CPU\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t",
      re,be,we,use,sye,ide,ie,ae,cse,l1e,l2e,l3e)
printf("%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t",
      r1,b1,w1,us1,sy1,id1,i1,a1,cs1,l11,l21,l31)
printf("%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t",
      r2,b2,w2,us2,sy2,id2,i2,a2,cs2,l12,l22,l32)
printf("%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t",
      r3,b3,w3,us3,sy3,id3,i3,a3,cs3,l13,l23,l33)
printf("%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t",
      r4,b4,w4,us4,sy4,id4,i4,a4,cs4,l14,l24,l34)
printf("DISQUE\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t",
      ops0,bps0,tps0,ops1,bps1,tps1,base,usr,var,tmp,users,students)
} ' /usr/users/inf/dedo/memoire/performance.jc >
/usr/users/inf/dedo/memoire/data/jour.dat
awk '
BEGIN{ avme=avm1=avm2=avm3=avm4=0
      freee=free1=free2=free3=free4=0
      ree=re1=re2=re3=re4=0
      pie=pi1=pi2=pi3=pi4=0
      poe=po1=po2=po3=po4=0
      fre=fr1=fr2=fr3=fr4=0
      reme=rem1=rem2=rem3=rem4=0

```

```

pime=pim1=pim2=pim3=pim4=0
inpe=inp1=inp2=inp3=inp4=0
inere=iner1=inere2=iner3=iner4=0
oute=out1=out2=out3=out4=0
outere=outer1=outer2=outer3=outer4=0
colle=coll1=coll2=coll3=coll4=0
}
{if ((NR>=49)&&(NR<=72))
    {avme=avme+$2
    freee=freee+$3
    avm1=avm1+$4
    free1=free1+$5
    avm2=avm2+$6
    free2=free2+$7
    avm3=avm3+$8
    free3=free3+$9
    avm4=avm4+$10
    free4=free4+$11}
if ((NR>=73)&&(NR<=96))
    {ree=ree+$2
    pie=pie+$3
    poe=poe+$4
    fre=fre+$5
    reme=reme+$6
    pime=pime+$7
    re1=re1+$8
    pi1=pi1+$9
    po1=po1+$10
    fr1=fr1+$11
    rem1=rem1+$12
    pim1=pim1+$13
    re2=re2+$14
    pi2=pi2+$15
    po2=po2+$16
    fr2=fr2+$17
    rem2=rem2+$18
    pim2=pim2+$19
    re3=re3+$20

```

```

pi3=pi3+$21
po3=po3+$22
fr3=fr3+$23
rem3=rem3+$24
pim3=pim3+$25
re4=re4+$26
pi4=pi4+$27
po4=po4+$28
fr4=fr4+$29
rem4=rem4+$30
pim4=pim4+$31}
if ((NR>=97)&&(NR<=120))
{inpe=inpe+$2
inere=inere+$3
oute=oute+$4
outere=outere+$5
colle=colle+$6
inp1=inp1+$7
iner1=iner1+$8
out1=out1+$9
outer1=outer1+$10
coll1=coll1+$11
inp2=inp2+$12
iner2=iner2+$13
out2=out2+$14
outer2=outer2+$15
coll2=coll2+$16
inp3=inp3+$17
iner3=iner3+$18
out3=out3+$19
outer3=outer3+$20
coll3=coll3+$21
inp4=inp4+$22
iner4=iner4+$23
out4=out4+$24
outer4=outer4+$25
coll4=coll4+$26}
}

```

END {avme=avme/24
 avm1=avm1/24
 avm2=avm2/24
 avm3=avm3/24
 freee=freee/24
 ree=ree/24
 pie=pie/24
 poe=poe/24
 fre=fre/24
 reme=reme/24
 pime=pime/24
 inpe=inpe/24
 inere=inere/24
 oute=oute/24
 outere=outere/24
 colle=colle/24
 free1=free1/24
 re1=re1/24
 pi1=pi1/24
 po1=po1/24
 fr1=fr1/24
 rem1=rem1/24
 pim1=pim1/24
 inp1=inp1/24
 iner1=iner1/24
 out1=out1/24
 outer1=outer1/24
 coll1=coll1/24
 free2=free2/24
 re2=re2/24
 pi2=pi2/24
 po2=po2/24
 fr2=fr2/24
 rem2=rem2/24
 pim2=pim2/24
 inp2=inp2/24
 iner2=iner2/24
 out2=out2/24

```

outer2=outer2/24
coll2=coll2/24
free3=free3/24
re3=re3/24
pi3=pi3/24
po3=po3/24
fr3=fr3/24
rem3=rem3/24
pim3=pim3/24
inp3=inp3/24
iner3=iner3/24
out3=out3/24
outer3=outer3/24
coll3=coll3/24
free4=free4/24
re4=re4/24
pi4=pi4/24
po4=po4/24
fr4=fr4/24
rem=rem4/24
pim4=pim4/24
inp4=inp4/24
iner4=iner4/24
out4=out4/24
outer4=outer4/24
coll4=coll4/24
printf("MEMOIRE\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\n",
      avme,freee,avm1,free1,avm2,free2,avm3,free3,avm4,free4)
printf("PAGE\t%s\t%s\t%s\t%s\t%s\t%s\t",ree,pie,poe,fe,reme,pime)
printf("%s\t%s\t%s\t%s\t%s\t%s\t",re1,pi1,po1,fr1,rem1,pim1)
printf("%s\t%s\t%s\t%s\t%s\t%s\t",re2,pi2,po2,fr2,rem2,pim2)
printf("%s\t%s\t%s\t%s\t%s\t%s\t",re3,pi3,po3,fr3,rem3,pim3)
printf("%s\t%s\t%s\t%s\t%s\t%s\t",re4,pi4,po4,fr4,rem4,pim4)
printf("RESEAU\t%s\t%s\t%s\t%s\t%s\t",inpe,inere,oute,outere,colle)
printf("%s\t%s\t%s\t%s\t%s\t",inp1,iner1,out1,outer1,coll1)
printf("%s\t%s\t%s\t%s\t%s\t",inp2,iner2,out2,outer2,coll2)
printf("%s\t%s\t%s\t%s\t%s\t",inp3,iner3,out3,outer3,coll3)
printf("%s\t%s\t%s\t%s\t%s\t",inp4,iner4,out4,outer4,coll4)

```

```

} ' /usr/users/inf/dedo/memoire/performance.jc >>
/usr/users/inf/dedo/memoire/data/jour.dat
cat /usr/users/inf/dedo/memoire/data/jour.dat
/usr/users/inf/dedo/memoire/performance.mc >
/usr/users/inf/dedo/memoire/data/aux.dat
awk '
BEGIN{d=2
      m=2
      p=2
      r=2
      i=0
      n=0}
{if (NR==1) {l1=$0}
 if (NR==2) {l2=$0}
 if (NR==3) {l3=$0}
 if (NR==4) {l4=$0}
 if (NR==5) {l5=$0}
 if ($1=="CPU") {n++}
 if (($1=="DISQUE")&&(d==0))
   {d=1
    print l1
    while (i<(31-n))
      {print " "
       i++}
    i=0}
 if (($1=="MEMOIRE")&&(m==0))
   {m=1
    print l2
    while (i<(31-n))
      {print " "
       i++}
    i=0}
 if (($1=="PAGE")&&(p==0))
   {p=1
    print l3
    while (i<(31-n))
      {print " "
       i++}

```

```

        i=0}
if (($1=="RESEAU")&&(r==0))
    {r=1
    print l4
    while (i<(31-n))
        {print " "
        i++}
    i=0}
if ((NR>5)&&(NF>1)) {print $0}
if (NR==5)
    {d=0
    m=0
    p=0
    r=0}
}
END{if (NR<6)
    {print l1
    n=NR-4
    while (i<(31-n))
        {print " "
        i=i+1}
    i=0
    print l2
    while (i<(31-n))
        {print " "
        i=i+1}
    i=0
    print l3
    while (i<(31-n))
        {print " "
        i=i+1}
    i=0
    print l4
    while (i<(31-n))
        {print " "
        i=i+1}
    i=0}
print l5

```



```

        while (i<(31-n))
            {print " "
             i=i+1}
    } ' /usr/users/inf/dedo/memoire/data/aux.dat >
    /usr/users/inf/dedo/memoire/performance.mc
cp /usr/users/inf/dedo/memoire/data/vidé.dat
    /usr/users/inf/dedo/memoire/performance.jc

```

F.13. MODULE COMPMOIS

```

echo 'compmois' >> /usr/users/inf/dedo/memoire/controle
date >> /usr/users/inf/dedo/memoire/controle
cp /usr/users/inf/dedo/memoire/performance.mc
    /usr/users/inf/dedo/memoire/performance.mp
date > /usr/users/inf/dedo/memoire/data/date.dat
cat /usr/users/inf/dedo/memoire/data/date.dat
    /usr/users/inf/dedo/memoire/performance.mc
    /usr/users/inf/dedo/memoire/archive >
    /usr/users/inf/dedo/memoire/data/aux3.dat
awk '
    {if (NR==1) {if ($2=="Jan") {print "Decembre",($6-1)}
                if ($2=="Feb") {print "Janvier", $6}
                if ($2=="Mar") {print "Fevrier", $6}
                if ($2=="Apr") {print "Mars", $6}
                if ($2=="May") {print "Avril", $6}
                if ($2=="Jun") {print "Mai", $6}
                if ($2=="Jul") {print "Juin", $6}
                if ($2=="Aug") {print "Juillet", $6}
                if ($2=="Sep") {print "Aout", $6}
                if ($2=="Oct") {print "Septembre", $6}
                if ($2=="Nov") {print "Octobre", $6}
                if ($2=="Dec") {print "Novembre", $6}}
    if (NR>=2) {print $0}
    } ' /usr/users/inf/dedo/memoire/data/aux3.dat >>
    /usr/users/inf/dedo/memoire/archive
cp /usr/users/inf/dedo/memoire/data/vidé.dat
    /usr/users/inf/dedo/memoire/performance.mc

```

F.14. MODULE COMBACKUP

```
echo 'compbackup' >> /usr/users/inf/dedo/memoire/controle
```

```
date >> /usr/users/inf/dedo/memoire/controle
```

```
cp /usr/users/inf/dedo/memoire/performance.jc
```

```
/usr/users/inf/dedo/memoire/data/auxb.dat
```

```
awk '
```

```
  {if ((NR>=1)&&(NR<9))
```

```
    {print $0}
```

```
  if ((NR>=9)&&(NR<13))
```

```
    {print "CPU  0    0    0    0    0    0    0    0
```

```
0    0    0    0    0    0    0    0    0    0
```

```
0    0    0    0    0    0    0    0    0    0
```

```
0    0    0    0    0    0    0    0    0    0
```

```
0    0    0    0    0    0    0    0    0    0
```

```
0    0    0    0    0    0    0    0"}}
```

```
  if ((NR>=13)&&(NR<33))
```

```
    {print $0}
```

```
  if ((NR>=33)&&(NR<37))
```

```
    {print "DISQUE  0    0    0    0    0    0    0
```

```
0    0    0    0    0"}}
```

```
  if ((NR>=37)&&(NR<57))
```

```
    {print $0}
```

```
  if ((NR>=57)&&(NR<61))
```

```
    {print "MEMOIRE  0    0    0    0    0    0    0
```

```
0    0    0    "}
```

```
  if ((NR>=61)&&(NR<81))
```

```
    {print $0}
```

```
  if ((NR>=81)&&(NR<85))
```

```
    {print "PAGE    0    0    0    0    0    0    0
```

```
0    0    0    0    0    0    0    0    0    0
```

```
0    0    0    0    0    0"}}
```

```
  if ((NR>=85)&&(NR<105))
```

```
    {print $0}
```

```
  if ((NR>=105)&&(NR<109))
```

```
    {print "RESEAU  0    0    0    0    0    0    0
```

```
0    0    0    0    0    0    0    0    0    0
```

```
0    0    0    0    0    0    0"}}
```

```
if ((NR>=109)&&(NR<=120))  
    {print $0}
```

```
} ' /usr/users/inf/dedo/memoire/data/auxb.dat >  
/usr/users/inf/dedo/memoire/performance.jc
```

ANNEXE G

LES FICHIERS WS*.DAT

G.1. FICHER WSUN.DAT

procs	memory	page	disk	faults	cpu
r b w	avm fre re at pi po fr de sr x0 x1 x2 x3 in sy	cs us sy id			
1 0 0	5688 4804 0 0 0 0 0 0 0 0 0 0 0 0 5 94	6 1 3 97			
1 0 0	5688 4804 2 8 1 0 0 0 0 0 0 0 0 0 38 205	26 2 2 96			
1 0 0	5688 4804 2 6 0 0 0 0 0 0 0 0 0 0 31 186	21 1 2 97			
1 0 0	5688 4804 2 4 0 0 0 0 0 0 0 0 0 0 25 171	17 1 2 97			
1 0 0	5688 4804 1 3 0 0 0 0 0 0 0 0 0 0 20 159	14 1 2 97			
1 0 0	5930 4541 1 2 0 0 0 0 0 0 0 0 0 0 16 147	11 2 2 96			
1 0 0	5930 4541 1 1 0 0 0 0 0 0 0 0 0 0 16 134	12 2 2 96			
1 0 0	5930 4541 1 0 0 0 0 0 0 0 0 0 0 0 13 129	10 1 2 97			
1 0 0	5930 4541 0 0 0 0 0 0 0 0 0 0 0 0 11 126	9 1 2 97			
1 0 0	5930 4541 0 0 0 0 0 0 0 0 0 0 0 0 9 123	8 1 2 97			

98086 reclaims, 1041305344 total time (usec)

average: 10616 usec / reclaim

4958 page ins, 104130534 total time (msec)

average: 21002.5 msec / page in

Name	Mtu	Network	Address	Ipkts	lerrs	Opkts	Oerrs	Coll
qe0	1500	138.48	ws1	350804	0	335353	0	5216
qe0	1500	DECnet	WS1	350804	0	335353	0	5216
lo0	1536	loop	localhost	7838	0	7838	0	0

28 swap ins

30 swap outs

926 pages swapped in

5978 pages swapped out

366126 total address trans. faults taken

4960 page ins

898 page outs

59744 pages paged in

20212 pages paged out

0 sequential process pages freed

98213 total reclaims (28% fast)

1902 reclaims from free list

5 intransit blocking page faults
 317617 zero fill pages created
 184110 zero fill page faults
 126893 executable fill pages created
 2985 executable fill page faults
 11 swap text pages found in free list
 68108 inode text pages found in free list
 0 file fill pages created
 0 file fill page faults
 21069 pages examined by the clock daemon
 3 revolutions of the clock hand
 4815 pages freed by the clock daemon
 1299279 cpu context switches
 21361068 device interrupts
 0 pseudo-dma dz interrupts
 377043 traps
 19053125 system calls
 11:43pm up 2 days, 8:13, 1 users, load average: 0.40, 0.13, 0.09
 User tty login@ idle JCPU PCPU what
 mophi console 8:35pm 61:52 evl

G.2. FICHER WSDEUX.DAT

procs		memory				page				disk faults				cpu						
r	b w	avm	fre	re	at	pi	po	fr	de	sr	x0	x1	x2	x3	in	sy	cs	us	sy	id
0	0	0	2108	1481	0	0	0	0	0	0	0	0	0	0	3	10	2	0	1	99
0	0	0	2108	1481	0	0	1	0	0	0	0	0	0	0	37	134	22	1	1	98
0	0	0	2108	1481	0	0	0	0	0	0	0	0	0	0	30	112	17	1	0	99
0	0	0	2108	1481	0	0	0	0	0	0	0	0	0	0	24	95	14	1	0	99
0	0	0	2458	1278	0	0	0	0	0	0	0	0	0	0	19	80	11	1	1	98
0	0	0	2458	1278	0	0	0	0	0	0	0	0	0	0	17	69	9	1	0	99
0	0	0	2458	1278	0	0	0	0	0	0	0	0	0	0	15	60	9	1	1	98
0	0	0	2458	1278	0	0	0	0	0	0	0	0	0	0	12	54	7	1	0	99
0	0	0	2458	1278	0	0	0	0	0	0	0	0	0	0	10	48	5	1	0	99
0	0	0	1838	1194	0	0	0	0	0	0	0	0	0	0	8	43	4	1	0	99

81388 reclaims, 1041305344 total time (usec)
 average: 12794 usec / reclaim
 4496 page ins, 104130534 total time (msec)

average: 23160.7 msec / page in

Name	Mtu	Network	Address	Ipkts	lerrs	Opkts	Oerrs	Coll
qe0	1500	138.48	ws2	667480	0	645969	14	5125
qe0	1500	DECnet	WS2	667480	0	645969	14	5125
lo0	1536	loop	localhost	14814	0	14814	0	0

9 swap ins

11 swap outs

94 pages swapped in

2115 pages swapped out

569698 total address trans. faults taken

4497 page ins

807 page outs

62764 pages paged in

18774 pages paged out

0 sequential process pages freed

81395 total reclaims (38% fast)

314 reclaims from free list

5 intransit blocking page faults

576959 zero fill pages created

334611 zero fill page faults

104786 executable fill pages created

3765 executable fill page faults

0 swap text pages found in free list

49492 inode text pages found in free list

0 file fill pages created

0 file fill page faults

15139 pages examined by the clock daemon

2 revolutions of the clock hand

3824 pages freed by the clock daemon

1120933 cpu context switches

40434661 device interrupts

0 pseudo-dma dz interrupts

582938 traps

4150465 system calls

11:45pm up 4 days, 12:07, 2 users, load average: 0.65, 0.22, 0.13

User	tty	login@	idle	JCPU	PCPU	what
------	-----	--------	------	------	------	------

pcr	ttyp0	9:01am	54:37	46	20	xman
-----	-------	--------	-------	----	----	------

pcr	ttyp1	3:55pm	36:36	16	16	rlogin electre
-----	-------	--------	-------	----	----	----------------

G.3. FICHIER WSTROIS.DAT

procs		memory		page		disk		faults		cpu											
r	b	w	avm	fre	re	at	pi	po	fr	de	sr	x0	x1	x2	x3	in	sy	cs	us	sy	id
1	0	0	2192	5493	0	0	0	0	0	0	0	0	0	0	0	4	16	4	1	1	98
1	0	0	2192	5493	2	9	1	0	0	0	0	0	0	0	0	38	135	22	1	0	99
1	0	0	2192	5493	1	7	0	0	0	0	0	0	0	0	0	30	112	17	1	0	99
1	0	0	2192	5493	1	5	0	0	0	0	0	0	0	0	0	24	94	13	1	0	99
1	0	0	2192	5493	0	4	0	0	0	0	0	0	0	0	0	19	80	10	1	0	99
0	0	0	2454	5216	0	3	0	0	0	0	0	0	0	0	0	15	68	8	1	1	98
0	0	0	2454	5216	0	2	0	0	0	0	0	0	0	0	0	15	60	8	1	1	98
0	0	0	2454	5216	0	1	0	0	0	0	0	0	0	0	0	12	53	7	1	0	99
0	0	0	2454	5216	0	0	0	0	0	0	0	0	0	0	0	10	47	5	1	0	99
0	0	0	2454	5216	0	0	0	0	0	0	0	0	0	0	0	8	42	4	1	0	99

100177 reclaims, 1041305344 total time (usec)
average: 10394 usec / reclaim

4732 page ins, 104130534 total time (msec)

average: 22005.6 msec / page in

Name	Mtu	Network	Address	lpkts	lerrs	Opkts	Oerrs	Coll
qe0	1500	138.48	ws3	386524	0	366123	1	4086
qe0	1500	DECnet	WS3	386524	0	366123	1	4086
lo0	1536	loop	localhost	8571	0	8571	0	0

10 swap ins

12 swap outs

179 pages swapped in

2692 pages swapped out

402334 total address trans. faults taken

4734 page ins

877 page outs

60716 pages paged in

19510 pages paged out

0 sequential process pages freed

100259 total reclaims (29% fast)

599 reclaims from free list

4 intransit blocking page faults

361511 zero fill pages created

209865 zero fill page faults

131149 executable fill pages created
 3646 executable fill page faults
 62 swap text pages found in free list
 70414 inode text pages found in free list
 0 file fill pages created
 0 file fill page faults
 16034 pages examined by the clock daemon
 3 revolutions of the clock hand
 3801 pages freed by the clock daemon
 886499 cpu context switches
 22720625 device interrupts
 0 pseudo-dma dz interrupts
 413109 traps
 3639309 system calls
 12:45am up 2 days, 12:29, 0 users, load average: 0.12, 0.04, 0.03
 User tty login@ idle JCPU PCPU what

G.4. FICHIER WSQUATRE.DAT

procs		memory		page		disk		faults		cpu										
r	b w	avm	fre	re	at	pi	po	fr	de	sr	x0	x1	x2	x3	in	sy	cs	us	sy	id
1	0 0	6042	4747	0	0	0	0	0	0	0	0	0	0	0	10	111	11	2	4	94
1	0 0	6042	4747	1	8	1	0	0	0	0	0	0	0	0	36	195	24	1	3	96
1	0 0	6042	4747	1	6	0	0	0	0	0	0	0	0	0	29	169	19	1	2	97
0	0 0	6562	4460	1	4	0	0	0	0	0	0	0	0	0	24	156	16	1	2	97
0	0 0	6562	4460	1	3	0	0	0	0	0	0	0	0	0	20	147	13	1	4	95
0	0 0	6562	4460	1	2	0	0	0	0	0	0	0	0	0	19	138	14	2	2	96
0	0 0	6562	4460	1	1	0	0	0	0	0	0	0	0	0	16	133	12	2	1	97
0	0 0	6562	4460	1	0	0	0	0	0	0	0	0	0	0	13	129	10	1	2	97
0	0 0	6562	4335	0	0	0	0	0	0	0	0	0	0	0	11	125	9	1	2	97
0	0 0	6562	4335	0	0	0	0	0	0	0	0	0	0	0	9	122	8	1	3	96

288423 reclaims, 1041305344 total time (usec)
 average: 3610 usec / reclaim

29541 page ins, 104130534 total time (msec)
 average: 3524.9 msec / page in

Name	Mtu	Network	Address	lpkts	lerrs	Opkts	Oerrs	Coll
------	-----	---------	---------	-------	-------	-------	-------	------

qe0	1500	138.48	ws4	819647	0	758497	1	8570
qe0	1500	DECnet	WS4	819647	0	758497	1	8570
lo0	1536	loop	localhost	14151	0	14151	0	0

246 swap ins
 250 swap outs
 5023 pages swapped in
 17598 pages swapped out
 924836 total address trans. faults taken
 29543 page ins
 6644 page outs
 289900 pages paged in
 137572 pages paged out
 0 sequential process pages freed
 288517 total reclaims (50% fast)
 15630 reclaims from free list
 277 intransit blocking page faults
 847253 zero fill pages created
 444069 zero fill page faults
 300800 executable fill pages created
 12990 executable fill page faults
 354 swap text pages found in free list
 127298 inode text pages found in free list
 0 file fill pages created
 0 file fill page faults
 154148 pages examined by the clock daemon
 26 revolutions of the clock hand
 39817 pages freed by the clock daemon
 3986848 cpu context switches
 39283293 device interrupts
 0 pseudo-dma dz interrupts
 972177 traps
 39705511 system calls
 11:43pm up 4 days, 2:39, 1 users, load average: 0.28, 0.10, 0.08
 User tty login@ idle JCPU PCPU what
 mophi console 7:52pm 62:04 evl

G.5. FICHER ELECTRE.DAT

procs	memory	page	disk	faults	cpu
r b w	avm fre re at pi po fr de sr r0 r1 x2 x3 in sy	cs us sy id			
5 0 0	2226 16k 0 0 0 0 0 0 0 0 0 1 2 0 0 18 25	12 1 3 96			
5 0 0	2226 16k 0 0 1 0 0 0 0 0 0 0 0 0 130 334	123 1 0 99			
5 0 0	2226 16k 0 0 0 0 0 0 0 0 0 0 0 0 104 272	98 0 0 100			
5 0 0	2226 16k 0 0 0 0 0 0 0 0 0 0 0 0 83 222	78 0 0 100			
0 0 0	2838 16k 0 0 0 0 0 0 0 0 1 5 0 0 66 182	62 0 6 94			
0 0 0	2838 16k 0 0 0 0 0 0 0 0 0 11 0 0 71 151	57 0 4 96			
0 0 0	2838 16k 0 0 0 0 0 0 0 0 0 1 0 0 58 125	46 0 1 99			
0 0 0	2838 16k 0 0 0 0 0 0 0 0 0 0 0 0 48 104	38 0 1 99			
0 0 0	2838 16k 0 0 0 0 0 0 0 9 0 0 0 0 38 88	30 0 2 98			
0 1 0	2848 16k 0 0 0 0 0 0 0 12 7 0 0 0 34 75	24 0 5 95			

17443 reclaims, 1041305344 total time (usec)

average: 59697 usec / reclaim

4060 page ins, 104130534 total time (msec)

average: 25647.9 msec / page in

tty	ra0	ra1	cpu
tin tout	bps tps	bps tps	us ni sy id
0 5	4 1 6 2	1 0 3 96	
0 0	8 1 52 16	17 0 54 29	
0 0	0 0 0 0	10 0 22 68	
0 0	0 0 0 0	33 0 46 21	
0 0	0 0 0 0	29 0 60 12	
0 0	7 1 49 15	13 0 34 53	
0 0	0 0 4 1	0 0 4 96	
0 0	0 0 0 0	30 0 49 22	
0 0	0 0 0 0	21 0 41 38	
0 0	0 0 0 0	0 0 0 6 94	

Filesystem Total kbytes kbytes %

node	kbytes	used	free	used	Mounted on
/dev/ra0a	7423	6466	215	97%	/
/dev/ra0e	187927	159248	9887	94%	/usr
/dev/ra0f	14383	2486	10459	19%	/var
/dev/ra0d	14383	14	12931	0%	/tmp
/dev/ra1g	177071	147310	12054	92%	/users
/dev/ra1a	28799	19767	6153	76%	/var/diskless/dlenv0

```

/dev/ra1b 162099 124616 21274 85% /var/diskless/dlclient0
/dev/ra0h 96399 81114 5646 93% /users/students
Name Mtu Network Address Ipkts lerrs Opkts Oerrs Coll
qe0 1500 DECnet ELEC 1284162 0 1305437 0 451
qe0 1500 138.48 electre 1284162 0 1305437 0 451
lo0 1536 loop localhost 182814 0 182814 0 0
0 swap ins
0 swap outs
4 pages swapped in
107 pages swapped out
522906 total address trans. faults taken
4068 page ins
0 page outs
32538 pages paged in
0 pages paged out
0 sequential process pages freed
17471 total reclaims (85% fast)
0 reclaims from free list
6 intransit blocking page faults
897644 zero fill pages created
336979 zero fill page faults
19445 executable fill pages created
3870 executable fill page faults
0 swap text pages found in free list
2588 inode text pages found in free list
0 file fill pages created
0 file fill page faults
0 pages examined by the clock daemon
0 revolutions of the clock hand
0 pages freed by the clock daemon
2528448 cpu context switches
24001433 device interrupts
0 pseudo-dma dz interrupts
579546 traps
5223956 system calls
11:03pm up 2 days, 8:21, 1 users, load average: 0.85, 0.26, 0.18
User tty login@ idle JCPU PCPU what
pcr ttyp0 3:12pm 36:36 1:58 7 -u

```

ANNEXE H

LES FICHIERS AUX*.DAT

H.1. FICHER AUX.DAT

CPU	1.01875	0.155208	0	0.96875	33.2563	61.5656
	397459	72165.3	36241.8	0.835833	0.338333	
	0.252604	0.165625	0.078125	0	1.04375	2.33125
	92.5542	362375	322328	21335.4	0.220208	
	0.119583	0.104063	0.14375	0.06875	0	0.897917
	0.472917	94.476	355739	28563.5	7697.21	
	0.194896	0.0913542	0.0719792	0.169792	0.0770833	0
	0.957292	0.411458	94.4635	355490	27411	
	7612.25	0.182708	0.0875	0.0721875	0.28125	
	0.0791667	0	1.2125	2.33958	92.2844	362365
	322315	21329.9	0.244479	0.126563	0.108438	
DISQUE	0.794792	5.05104	0.76875	1.55	3.75938	
	1.15625	92.9583	90.0833	17.875	0	88.1667
	89.125					
MEMOIRE	2484.04	15333.3	5516.33	4501.58	1906.76	
	1283.32	1934.62	5102.85	147129	4328.66	
PAGE	0	375.667	0	0	66090.3	29510.5
	0.166667	271.833	0	0	12113.1	21151.7
	267.75	0	0	12507.3	23402	0
	0	0	11809.9	22173.7	0	268.333
	3650.2	3404.37				
RESEAU	19394.3	0	19417	0	1.66667	5400.67
	0	5256.42	0	23.2917	5464.92	0
	0	12.625	5336.96	0	5194.79	0
	5395.42	0	5252.38	0	26.2917	
CPU	4.61146	0.601042	30.3167	2.80417	18.6948	
	78.5083	657723	77856	486840	1.81708	
	1.24417	1.13771	0.284375	0.205208	0	2.20208
	3.02083	94.4667	389289	374993	29231.1	
	0.401146	0.212292	0.182292	0.141667	0.125	0
	0.896875	0.470833	98.6323	371512	29021.3	
	8404.92	0.205729	0.0977083	0.0788542	0.176042	
	0.201042	0	1.43229	0.555208	97.9937	374859

	48164.1	13003.7	0.323021	0.194479	0.174688	
	1.29688	0.2625	0	23.2625	53.4469	23.4708
	609901	1.36369e+06	258104	1.5049	1.33021	
	1.30146					
CPU	1229.88	10.9073	2.1694e+06	2.21667	21.1406	
	58.7323	451728	67861.3	29964082	1.15708	
	0.726458	0.651875	0.241667	54.4083	0	1120.47
	1779.49	1681.24	317741	315614	27688.1	
	0.33875	0.199792	0.178438	4.23333	1735.14	
	1.35417	445114	29.9417	94.5927	296152	
	31967.9	8834.33	0.18875	0.0921875	0.07625	
	0.111458	0.107292	0	1.79479	1.15521	75.9
	312115	129713	28175.5	0.238958	0.138333	
	0.11875	0.63125	0.247917	0	9.67813	21.001
	47.8198	393453	675047	111377	0.773958	0.615
	0.575					
CPU	2.20729	0.782292	9.53125	1.7375	29.1365	
	69.1042	456475	86666.7	71570	1.11646	
	0.550208	0.453854	0.289583	0.179167	0	3.67188
	3.71563	92.5781	402720	398530	35137.2	
	0.370833	0.222083	0.2	0.164583	0.157292	0
	1.00521	0.452083	98.5427	372918	35715.3	
	9646.29	0.229896	0.11	0.0878125	2.75417	0.209375
	1.08469e+06	2.80208	1.8	95.001	380061	
	171711	40725.3	0.366563	0.243958	0.219896	
	0.355208	0.215625	0.00208333	4.80208	3.98125	
	91.2156	418319	417880	42397.5	0.455833	
	0.287292	0.258021				
CPU	1.65625	0.654167	1.08469e+06	1.90625	24.1865	
	70.6854	449453	134288	73132.5	1.08719	
	0.570313	0.483021	3.51979	0.15	1.08469e+06	
	3.02396	2.83021	89.7312	384285	354062	
	38824.5	0.297083	0.190833	0.175	75.2706	0.109375
	1.08469e+06	1.21146	1.01771	94.4771	360428	
	43286.2	823744	0.246458	0.117187	0.129583	
	1.21875	0.190625	1.08469e+06	2.19375	1.52188	
	91.4938	436839	176292	44320.3	0.27875	
	0.155521	0.160521	0.290625	0.198958	0	5.84479

	4.18333	86.3219	1.13132e+06	1.18136e+06	
	144337	42.6451	0.300417	35.0323	
CPU	1.04271	0.0697917	0	1.02708	32.7823 66.1927
	414936	75732	38004.6	0.8525	0.348438
	0.257917	0.257292	0.0770833	0	1.10208 2.52187
	96.3802	378111	336260	22251.4	0.242292
	0.127396	0.109375	0.135417	0.0989583	0 1.00313
	0.419792	98.5771	371924	31926.8	8565.75 0.2
	0.0948958	0.0755208	0.184375	0.109375	0 1.02188
	0.519792	98.4583	370934	28612.8	7950.08
	0.224896	0.104063	0.0819792	0.19375	0.0614583 0
	1.31979	2.50417	96.1698	378117	336314
	22264.1	0.220521	0.11375	0.0992708	

.
.
.

DISQUE	1.39687	9.225	1.46979	4.48229	12.7417
	3.60938	97	94	30.4375	0 92 65
DISQUE	78.9573	39.175	13.049	12.8719	50.0958
	47.2969	78.7813	75.7083	24.4896	0.958333
	74.5833	52.875			
DISQUE	1.73542	12.3542	1.97083	5.7125	15.1979
	4.28438	97	94	30.5417	0 92 67.6771
DISQUE	2.10417	14.0531	2.2125	5.59375	16.7104
	4.63646	93.9375	90.4063	26.6458	0.96875 89
	74.1563				
DISQUE	0.780208	6.35	0.938542	1.4875	6.07708
	1.84063	97	94	18.9479	0 92 93

.
.
.

MEMOIRE	3329.85	14875	6950.34	3286.8	1847.84
	4338.02	2883.07	5085.26	146780	3100.58

MEMOIRE	2737.27	13326.6	5894.48	2290.06	202931
	3720.62	2654.11	3791.29	118197	3045.97
MEMOIRE	3524.56	14712.5	7260.05	3488.34	1944.15
	4996.25	2807.01	5093.78	168240	3673.8
MEMOIRE	3154.53	13829.5	6132.31	3552.87	1899.35
	2694.35	2373.93	4524.22	158024	3716.77
MEMOIRE	2541.24	16000	5716.73	4738.3	1755.01
	1338.3	1803.63	5327.83	151386	4521.56

.
.
.

PAGE	0	392.583	0	0	1409.1	2493.39
	154.792	2771.5	1419.33	660.542	4743.24	
	5903.93	0.125	370.083	0	0	10558.2
	4.5	712.75	262.5	65.2917	47114.3	65701.3
	86.4583	2905.75	1048	468.375	3217.29	1911.95

PAGE	45384.6	1832.75	4.38051e+06	6.03084e+06		
	1154.8	1936.5	1803.47	4248.24	8679613	
	828.675	2722.76	2889.7	2.875	651.688	163.417
	27.1417	33954.9	49709.4	38.7083	2207.92	
	1232.17	294.5	54075.9	67773.8	143.708	4424.08
	2133.25	581.542	49486.5	46258.7		

PAGE	0	526.833	0	0	1276.07	2333.41
	222.667	4292.08	1900.5	819.333	2626.22	
	2415.14	2.08333	580.75	144.583	27.625	
	25940.1	52713.6	676.25	3479.28	1228.92	
	1099.96	47306.9	57237.3	203.458	4094.25	
	1937.25	563.958	32346.4	42143.5		

PAGE	0.0833333	935.083	0	1153	85362.4	56417.6
	255.75	4913.75	2304.5	1221.92	16352.7	
	12861.8	3.66667	1142.25	276.333	25649.2	
	14486.6	32654.5	917.708	2523.79	1770.29	
	484.458	43023.9	50786.6	5078.17	15193.7	
	10086.4	2097.21	7068.78	6686.33		

PAGE	0	422.083	0	0	101672	52019.2
	0.0833333	281	0	0	20356.6	24561.7
						0.0416667

281.5	4.08333	2.75	13618.7	27492.5	0	280.167
0	0	19531.9	25842.7	0.0833333	281.083	0
0	4289.38	3610.44				

RESEAU	137763	0	138956	0	5.04167	8541.83
0	7852.79	0	11.125	5885.88	0	5664.67
0	7.58333	6520.08	0	6168.25	0	10.125
120412	0	119764	0	7.5		
RESEAU	194380	190.579	196843	0.116667	587.433	
9187.98	0	7863	366.208	92.625	5073.92	0
4787.83	0	7.125	7114.46	0	6442.92	0
8.29167	49923.1	0	49027.2	0	11.9167	
RESEAU	32447.2	0.0416667	35756	0	8.91667	
10379.5	0	9237.08	0	22.5	6308.38	0
6008.33	0	9.125	1766.13	0	1413.13	0
13.3333	10935.2	0	9656.71	0	20.25	
RESEAU	32429.6	0	34872.4	0	3.79167	9904.83
0	8955.04	0.166667	18	20287.3	0	19593.9
1.16667	8.95833	2100.92	0	1973.33	0.0416667	
20.9167	28401.7	0	25974.6	0.0833333	158.25	
RESEAU	20330.5	0	20338.7	0	1.5	5628.33
5478.5	0	24.2917	5763.42	0	5634.42	0
15.9167	5574.29	0	5423.25	0	17.0833	
5630.21	0	5483.17	0	25.5		

H.2. FICHER AUXUN.DAT

349377 0 333958 0 5185 59674 20212 1902
 4815 1293654 21266377 18968857

procs	memory	page	disk faults	cpu
r b w	avm	fre	re at pi po fr de sr x0 x1 x2 x3 in sy cs us sy id	
1 0 0	5688	4804	0 0 0 0 0 0 0 0 0 0 0 0 0 5 94	6 1 3 97
1 0 0	5688	4804	2 8 1 0 0 0 0 0 0 0 0 0 0 38 205	26 2 2 96
1 0 0	5688	4804	2 6 0 0 0 0 0 0 0 0 0 0 0 31 186	21 1 2 97
1 0 0	5688	4804	2 4 0 0 0 0 0 0 0 0 0 0 0 25 171	17 1 2 97
1 0 0	5688	4804	1 3 0 0 0 0 0 0 0 0 0 0 0 20 159	14 1 2 97
1 0 0	5930	4541	1 2 0 0 0 0 0 0 0 0 0 0 0 16 147	11 2 2 96
1 0 0	5930	4541	1 1 0 0 0 0 0 0 0 0 0 0 0 16 134	12 2 2 96
1 0 0	5930	4541	1 0 0 0 0 0 0 0 0 0 0 0 0 13 129	10 1 2 97
1 0 0	5930	4541	0 0 0 0 0 0 0 0 0 0 0 0 0 11 126	9 1 2 97
1 0 0	5930	4541	0 0 0 0 0 0 0 0 0 0 0 0 0 9 123	8 1 2 97

98086 reclaims, 1041305344 total time (usec)

average: 10616 usec / reclaim

4958 page ins, 104130534 total time (msec)

average: 21002.5 msec / page in

Name	Mtu	Network	Address	Ipkts	lerrs	Opkts	Oerrs	Coll
qe0	1500	138.48	ws1	350804	0	335353	0	5216
qe0	1500	DECnet	WS1	350804	0	335353	0	5216
lo0	1536	loop	localhost	7838	0	7838	0	0

28 swap ins

30 swap outs

926 pages swapped in

5978 pages swapped out

366126 total address trans. faults taken

4960 page ins

898 page outs

59744 pages paged in

20212 pages paged out

0 sequential process pages freed

98213 total reclaims (28% fast)

1902 reclaims from free list

5 intransit blocking page faults

317617 zero fill pages created

184110 zero fill page faults
 126893 executable fill pages created
 2985 executable fill page faults
 11 swap text pages found in free list
 68108 inode text pages found in free list
 0 file fill pages created
 0 file fill page faults
 21069 pages examined by the clock daemon
 3 revolutions of the clock hand
 4815 pages freed by the clock daemon
 1299279 cpu context switches
 21361068 device interrupts
 0 pseudo-dma dz interrupts
 377043 traps
 19053125 system calls
 11:43pm up 2 days, 8:13, 1 users, load average: 0.40, 0.13, 0.09
 User tty login@ idle JCPU PCPU what
 mophi console 8:35pm 61:52 evl

H.3. FICHER AUXDEUX.DAT

666048 0 644568 14 5112 62708 18774 314
 3824 1118897 40341446 4143252
 procs memory page disk faults cpu
 r b w avm fre re at pi po fr de sr x0 x1 x2 x3 in sy cs us sy id
 0 0 0 2108 1481 0 0 0 0 0 0 0 0 0 0 0 3 10 2 0 1 99
 0 0 0 2108 1481 0 0 1 0 0 0 0 0 0 0 0 37 134 22 1 1 98
 0 0 0 2108 1481 0 0 0 0 0 0 0 0 0 0 0 30 112 17 1 0 99
 0 0 0 2108 1481 0 0 0 0 0 0 0 0 0 0 0 24 95 14 1 0 99
 0 0 0 2458 1278 0 0 0 0 0 0 0 0 0 0 0 19 80 11 1 1 98
 0 0 0 2458 1278 0 0 0 0 0 0 0 0 0 0 0 17 69 9 1 0 99
 0 0 0 2458 1278 0 0 0 0 0 0 0 0 0 0 0 15 60 9 1 1 98
 0 0 0 2458 1278 0 0 0 0 0 0 0 0 0 0 0 12 54 7 1 0 99
 0 0 0 2458 1278 0 0 0 0 0 0 0 0 0 0 0 10 48 5 1 0 99
 0 0 0 1838 1194 0 0 0 0 0 0 0 0 0 0 0 8 43 4 1 0 99
 81388 reclaims, 1041305344 total time (usec)
 average: 12794 usec / reclaim
 4496 page ins, 104130534 total time (msec)

average: 23160.7 msec / page in

Name	Mtu	Network	Address	Ipkts	lerrs	Opkts	Oerrs	Coll
qe0	1500	138.48	ws2	667480	0	645969	14	5125
qe0	1500	DECnet	WS2	667480	0	645969	14	5125
lo0	1536	loop	localhost	14814	0	14814	0	0

9 swap ins

11 swap outs

94 pages swapped in

2115 pages swapped out

569698 total address trans. faults taken

4497 page ins

807 page outs

62764 pages paged in

18774 pages paged out

0 sequential process pages freed

81395 total reclaims (38% fast)

314 reclaims from free list

5 intransit blocking page faults

576959 zero fill pages created

334611 zero fill page faults

104786 executable fill pages created

3765 executable fill page faults

0 swap text pages found in free list

49492 inode text pages found in free list

0 file fill pages created

0 file fill page faults

15139 pages examined by the clock daemon

2 revolutions of the clock hand

3824 pages freed by the clock daemon

1120933 cpu context switches

40434661 device interrupts

0 pseudo-dma dz interrupts

582938 traps

4150465 system calls

11:45pm up 4 days, 12:07, 2 users, load average: 0.65, 0.22, 0.13

User	tty	login@	idle	JCPU	PCPU	what
------	-----	--------	------	------	------	------

pcr	ttyp0	9:01am	54:37	46	20	xman
-----	-------	--------	-------	----	----	------

pcr	ttyp1	3:55pm	36:36	16	16	rlogin electre
-----	-------	--------	-------	----	----	----------------

H.4. FICHER AUXTROIS.DAT

385146 0 364779 1 4062 60646 19510 599

3801 884524 22627920 3632471

procs		memory		page		disk		faults		cpu	
r	b w	avm	fre	re	at	pi	po	fr	de	sr	x0 x1 x2 x3 in sy cs us sy id
1	0 0	2192	5493	0	0	0	0	0	0	0	0 0 0 0 4 16 4 1 1 98
1	0 0	2192	5493	2	9	1	0	0	0	0	0 0 0 0 38 135 22 1 0 99
1	0 0	2192	5493	1	7	0	0	0	0	0	0 0 0 0 30 112 17 1 0 99
1	0 0	2192	5493	1	5	0	0	0	0	0	0 0 0 0 24 94 13 1 0 99
1	0 0	2192	5493	0	4	0	0	0	0	0	0 0 0 0 19 80 10 1 0 99
0	0 0	2454	5216	0	3	0	0	0	0	0	0 0 0 0 15 68 8 1 1 98
0	0 0	2454	5216	0	2	0	0	0	0	0	0 0 0 0 15 60 8 1 1 98
0	0 0	2454	5216	0	1	0	0	0	0	0	0 0 0 0 12 53 7 1 0 99
0	0 0	2454	5216	0	0	0	0	0	0	0	0 0 0 0 10 47 5 1 0 99
0	0 0	2454	5216	0	0	0	0	0	0	0	0 0 0 0 8 42 4 1 0 99

100177 reclaims, 1041305344 total time (usec)

average: 10394 usec / reclaim

4732 page ins, 104130534 total time (msec)

average: 22005.6 msec / page in

Name	Mtu	Network	Address	lpkts	lerrs	Opkts	Oerrs	Coll
qe0	1500	138.48	ws3	386524	0	366123	1	4086
qe0	1500	DECnet	WS3	386524	0	366123	1	4086
lo0	1536	loop	localhost	8571	0	8571	0	0

10 swap ins

12 swap outs

179 pages swapped in

2692 pages swapped out

402334 total address trans. faults taken

4734 page ins

877 page outs

60716 pages paged in

19510 pages paged out

0 sequential process pages freed

100259 total reclaims (29% fast)

599 reclaims from free list

4 intransit blocking page faults

361511 zero fill pages created

209865 zero fill page faults
 131149 executable fill pages created
 3646 executable fill page faults
 62 swap text pages found in free list
 70414 inode text pages found in free list
 0 file fill pages created
 0 file fill page faults
 16034 pages examined by the clock daemon
 3 revolutions of the clock hand
 3801 pages freed by the clock daemon
 886499 cpu context switches
 22720625 device interrupts
 0 pseudo-dma dz interrupts
 413109 traps
 3639309 system calls
 12:45am up 2 days, 12:29, 0 users, load average: 0.12, 0.04, 0.03
 User tty login@ idle JCPU PCPU what

H.5. FICHER AUXQUATRE.DAT

818246 0 757128 1 8535 289830 137572 15630
 39817 3981299 39188890 39621544
 procs memory page disk faults cpu
 r b w avm fre re at pi po fr de sr x0 x1 x2 x3 in sy cs us sy id
 1 0 0 6042 4747 0 0 0 0 0 0 0 0 0 0 10 11 11 2 4 94
 1 0 0 6042 4747 1 8 1 0 0 0 0 0 0 0 36 195 24 1 3 96
 1 0 0 6042 4747 1 6 0 0 0 0 0 0 0 0 29 169 19 1 2 97
 0 0 0 6562 4460 1 4 0 0 0 0 0 0 0 0 24 156 16 1 2 97
 0 0 0 6562 4460 1 3 0 0 0 0 0 0 0 0 20 147 13 1 4 95
 0 0 0 6562 4460 1 2 0 0 0 0 0 0 0 0 19 138 14 2 2 96
 0 0 0 6562 4460 1 1 0 0 0 0 0 0 0 0 16 133 12 2 1 97
 0 0 0 6562 4460 1 0 0 0 0 0 0 0 0 0 13 129 10 1 2 97
 0 0 0 6562 4335 0 0 0 0 0 0 0 0 0 0 11 125 9 1 2 97
 0 0 0 6562 4335 0 0 0 0 0 0 0 0 0 0 9 122 8 1 3 96
 288423 reclaims, 1041305344 total time (usec)
 average: 3610 usec / reclaim

29541 page ins, 104130534 total time (msec)

average: 3524.9 msec / page in

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
qe0	1500	138.48	ws4	819647	0	758497	1	8570
qe0	1500	DECnet	WS4	819647	0	758497	1	8570
lo0	1536	loop	localhost	14151	0	14151	0	0

246 swap ins

250 swap outs

5023 pages swapped in

17598 pages swapped out

924836 total address trans. faults taken

29543 page ins

6644 page outs

289900 pages paged in

137572 pages paged out

0 sequential process pages freed

288517 total reclaims (50% fast)

15630 reclaims from free list

277 intransit blocking page faults

847253 zero fill pages created

444069 zero fill page faults

300800 executable fill pages created

12990 executable fill page faults

354 swap text pages found in free list

127298 inode text pages found in free list

0 file fill pages created

0 file fill page faults

154148 pages examined by the clock daemon

26 revolutions of the clock hand

39817 pages freed by the clock daemon

3986848 cpu context switches

39283293 device interrupts

0 pseudo-dma dz interrupts

972177 traps

39705511 system calls

11:43pm up 4 days, 2:39, 1 users, load average: 0.28, 0.10, 0.08

User tty login@ idle JCPU PCPU what

mophi console 7:52pm 62:04 evl

H.6. FICHER AUXE.DAT

```
1279058    0    1300336    0    450  32440    0    0    0
      2519000    23897421  5204999
```

```
procs  memory          page  disk faults    cpu
r b w  avm fre re at pi po fr de sr r0 r1 x2 x3 in sy cs us sy id
5 0 0 2226 16k 0 0 0 0 0 0 0 0 1 2 0 0 18 25 12 1 3 96
5 0 0 2226 16k 0 0 1 0 0 0 0 0 0 0 0 0 130 334 123 1 0 99
5 0 0 2226 16k 0 0 0 0 0 0 0 0 0 0 0 0 104 272 98 0 0 100
5 0 0 2226 16k 0 0 0 0 0 0 0 0 0 0 0 0 83 222 78 0 0 100
0 0 0 2838 16k 0 0 0 0 0 0 0 0 1 5 0 0 66 182 62 0 6 94
0 0 0 2838 16k 0 0 0 0 0 0 0 0 0 11 0 0 71 151 57 0 4 96
0 0 0 2838 16k 0 0 0 0 0 0 0 0 0 1 0 0 58 125 46 0 1 99
0 0 0 2838 16k 0 0 0 0 0 0 0 0 0 0 0 0 48 104 38 0 1 99
0 0 0 2838 16k 0 0 0 0 0 0 0 0 9 0 0 0 38 88 30 0 2 98
0 1 0 2848 16k 0 0 0 0 0 0 0 0 12 7 0 0 34 75 24 0 5 95
```

17443 reclaims, 1041305344 total time (usec)

average: 59697 usec / reclaim

4060 page ins, 104130534 total time (msec)

average: 25647.9 msec / page in

```
tty    ra0    ra1    cpu
tin tout bps tps bps tps us ni sy id
0 5 4 1 6 2 1 0 3 96
0 0 8 1 52 16 17 0 54 29
0 0 0 0 0 0 10 0 22 68
0 0 0 0 0 0 33 0 46 21
0 0 0 0 0 0 29 0 60 12
0 0 7 1 49 15 13 0 34 53
0 0 0 0 4 1 0 0 4 96
0 0 0 0 0 0 30 0 49 22
0 0 0 0 0 0 21 0 41 38
0 0 0 0 0 0 0 0 6 94
```

Filesystem Total kbytes kbytes %

```
node    kbytes used free used Mounted on
/dev/ra0a 7423 6466 215 97% /
/dev/ra0e 187927 159248 9887 94% /usr
/dev/ra0f 14383 2486 10459 19% /var
/dev/ra0d 14383 14 12931 0% /tmp
```

```

/dev/ra1g  177071 147310 12054 92% /users
/dev/ra1a  28799 19767 6153 76% /var/diskless/dlenv0
/dev/ra1b  162099 124616 21274 85% /var/diskless/dlclient0
/dev/ra0h  96399 81114 5646 93% /users/students
Name Mtu Network Address Ipkts lerrs Opkts Oerrs Coll
qe0 1500 DECnet ELEC 1284162 0 1305437 0 451
qe0 1500 138.48 electre 1284162 0 1305437 0 451
lo0 1536 loop localhost 182814 0 182814 0 0
0 swap ins
0 swap outs
4 pages swapped in
107 pages swapped out
522906 total address trans. faults taken
4068 page ins
0 page outs
32538 pages paged in
0 pages paged out
0 sequential process pages freed
17471 total reclaims (85% fast)
0 reclaims from free list
6 intransit blocking page faults
897644 zero fill pages created
336979 zero fill page faults
19445 executable fill pages created
3870 executable fill page faults
0 swap text pages found in free list
2588 inode text pages found in free list
0 file fill pages created
0 file fill page faults
0 pages examined by the clock daemon
0 revolutions of the clock hand
0 pages freed by the clock daemon
2528448 cpu context switches
24001433 device interrupts
0 pseudo-dma dz interrupts
579546 traps
5223956 system calls
11:03pm up 2 days, 8:21, 1 users, load average: 0.85, 0.26, 0.18

```


User tty login@ idle JCPU PCPU what
 pcr tty0 3:12pm 36:36 1:58 7 -u

H.7. FICHER AUX2.DAT

CPU	1.15	0.125	0	0.6	22.225	77.125	414792	75857
	37493		0.9375		0.3525	0.245 0.35	0 0	1.125
	2.175	96.8	378124		336546	22269	0.34 0.145	0.115
	0.25	0	0	0.9	0.5 98.6	371156	29443	7996
	0.3425		0.1325		0.0825	0.25 0 0	1.025	0.3
	98.675		370694		28272	7899 0.1225	0.055	0.05
	0.15	0	0	1.225	2.35 96.425	378296	336464	
	22236		0.2525		0.105 0.085			
DISQUE	0.675	3.325	0.6		1.275 5.75	1.75 97 94	19	0
	92	93						
MEMOIRE	2612.55		16000		5775.95	4667.9	2189.95	
	1344.55		2181.95		5331.5	6416.65	4518.25	
PAGE	0	392	0	0	59956.3	25784.8	0	280
	0	0	10670.3		21034.4	0 266 0 0		12802
	23199.4		0 280	0 0	10447.5	22041.8	0	
	294	0	0 3616	3525.8				
RESEAU	20175		0	20165	0 1	5612 0	5471	
	0 31	5668	0 5533	0 13	5537 0	5396 0		
	24 5596	0 5457	0 35					
CPU	0.475	0.15	0	2.625	28.95 68.4	414289	74717	38060
	0.7875		0.2975		0.2375	0.125 0.125 0	1.075	2.95
	96.075		377793		336056	22231	0.25 0.1475	
	0.135	0.125	0.075 0		0.975 0.55	98.475	370990	29745
	8006	0.2075		0.075	0.055 0.325	0.125 0 1	0.525	98.45
	371112		28542		7897 0.2075	0.125 0.11 0.25	0	
	0 1.45	2.35	96.2	378012	336516	22258		
	0.3075		0.12 0.1025					
CPU	1.425	0.125	0	1.6	28.4 70	414356	74508	37490
	0.995	0.3575		0.225 0	0.1 0	1.1 2.475	96.525	
	378508		336410		22258	0.225 0.1175	0.095	0.35
	0.25	0	1	0.475	98.525	371324	29757	8027
	0.2275		0.1025		0.08 0.2	0 0 0.975	0.55	
	98.475		371053		28530	7942 0.1925	0.0775	

	0.0575	0.525	0.125	0	1.425	2.525	96.05	377932	
	336042	22255		0.265	0.145	0.1275			
CPU	1.275	0.425	0	0.95	25.05	74	415605	76424	38917
	1.095	0.4275		0.285	0.15	0.25	0	1.075	2.8
	377599	335965		22219			0.255	0.125	0.11
	0	1	0.425	98.575		371092	29554	8013	0.235
	0.0925	0.0675		0.375	0	0	1	0.45	98.55
	371416	29791		8417	0.23	0.1025		0.08	0.5
	0	1.3	2.6	96.1	378416		336662	22216	
	0.2975		0.13	0.1025					
CPU	0.975	0.1	0	0.175	24.95	74.875		418479	79043
	41642		0.7375		0.295	0.2125		0.375	0.2
	2.15	96.825		379294		337639		22798	0.3125
	0.1725		0.15	0	0.25	0	0.95	0.575	98.475
	371974		31180		8463	0.3325		0.115	0.08
	0	1	0.475	98.525		371066		28831	7936
	0.0975		0.0925		0.35	0	0	1.35	2.4
	378992		337538		22802		0.21	0.0975	0.09
CPU	1.275	0.2	0	0.25	51.875		47.875	414944	77533
	37994		0.785	0.3375		0.2575		0.2	0
	2.325	96.625		377819		335774		22263	0.245
	0.115	0	0	0	1	0.45	98.55	371104	29764
	8007	0.14	0.08	0.0675		0.3	0.05	0	1
	370926		28557		7919	0.2075		0.1	0.09
	0	1.4	2.35	96.25	378000		336086	22242	0.18
	0.1075		0.0925						
CPU	0.575	0	0	3.15	48.65	48.225		414168	74858
	37246		1.1225		0.445	0.3325		0.05	0.15
	2.375	96.625		378087		336150		22196	0.2125
	0.12	0.1125		0.125	0.125	0		0.975	0.55
	371220		29857		8021	0.14	0.095	0.07	0.15
	1.05	0.475	98.475		370829		28529		7918
	0.075	0.0725		0.5	0	0		1.375	2.425
	378110		336160		22248		0.245	0.1275	0.1
CPU	0.95	0	0	0.4	55.025		44.575	415004	77396
	37972		0.8675		0.3375		0.255	0.25	0
	2.975	96.15	378414		336851		22274		0.21
	0.105	0	0.125	0	0.975	0.375	98.65	371320	29955

	8037	0.265	0.12	0.095	0.125	0	0	1	0.675	98.325	
	371133		28577		7932	0.12	0.0725		0.065	0.45	0.125
	0	1.225	2.5	96.275		377896		335920		22221	
	0.2675		0.135	0.1125							
CPU	1.325	0.5	0	0.15	36.625		63.225		414644		76495
	37539		0.855	0.3775		0.2775		0.15	0.3	0	1.075
	2.35	96.675		377668		335584		22191		0.2575	
	0.1275		0.11	0.3	0	0	1	0.55	98.525		
	371084		29643		8004	0.2575		0.1025		0.09	0.1
	0.3	0	1	0.4	98.6	370824		28551		7940	0.145
	0.0725		0.07	0.25	0.25	0	1.175	2.375	96.45	378200	
	336188		22256		0.2325		0.11	0.1			
CPU	0.875	0	0	1.525	32.125		66.35	414562		73953	
	37617		0.7175		0.34	0.295	0.1	0	0	1.075	2.275
	96.75	378295		336487		22228		0.1025		0.085	
	0.0825		0.25	0	0	1	0.325	98.775		371076	
	29854		7999	0.15	0.075	0.06	0.25	0	0	1.05	0.375
	98.575		370782		28246		7869	0.205	0.0875		
	0.0725		0.25	0	0	1.175	2.3	96.525		377972	
	336188		22227		0.2425		0.1375		0.115		
CPU	1.025	0.25	0	0.225	17.675		82.1	414912		77127	
	37896		0.8825		0.355	0.26	0.175	0	0	1.075	2.7
	96.325		378224		336425		22264		0.2025		0.11
	0.09	0.35	0	0	1	0.675	98.425		371404		29565
	8027	0.27	0.115	0.09	0	0.1	0	1.025	0.525	98.45	
	371182		28842		7970	0.13	0.065	0.05	0.375	0.3	0
	1.15	3.075	95.775		378440		336724		22257		
	0.2825		0.1725		0.14						
CPU	0.725	0.175	0	1.175	36.725		62.1	414058		75486	
	37423		0.6475		0.2975		0.235	0.125	0	0	1.1
	2.225	96.775		378193		336751		22209		0.1625	
	0.1025		0.0975		0.125	0	0	0.95	0.475	98.625	
	370888		29752		7995	0.175	0.1	0.0825		0.05	0
	0	1	0.225	98.775		370906		28549		7908	
	0.1625		0.0775		0.0625		0.375	0	0	1.225	2.2
	96.575		377948		335748		22209		0.29	0.19	0.17
CPU	1.75	0.125	0	0.475	30.475		69.05	414178		75748	
	38164		0.7875		0.3675		0.2825		0.05	0	0

	1.1	2.125	96.875		377914	336201	22228		0.13		
	0.075	0.07	0.25	0.1	0	0.9	0.5	98.6	371304	29696	
	8019	0.23	0.1325		0.095	0.075	0	0	0.95	0.325	
	98.725		370458		28560		7943	0.195	0.0775		
	0.0525		0.4	0.025	0	1.25	2.2	96.55	377904		
	336332		22247		0.2175		0.115	0.1025			
CPU 1	0	0		1.075	50.075		48.75	414936		73882	
	37294		0.69	0.2775		0.2225		0.125	0.125	0	1.125
	2.45	96.525		378276		336693		22264		0.165	0.1
	0.0925		0.05	0.275	0	0.85	0.525	98.625		371168	
	29657		8011	0.1475		0.105	0.095	0.075	0.125	0	0.975
	0.575	98.45	371208		28360		7912	0.1225		0.065	0.06
	0.3	0	0	1.3	2.45	96.25	377804		336160		22238
	0.3525		0.14	0.11							
CPU 1.3	0	0		2.075	24.325		73.475		414674		74431
	37149		0.975	0.4	0.275	0.1	0	0	1.1	2.425	
	96.575		378132		336086		22247		0.1775		0.12
	0.1025		0.025	0	0	0.9	0.4	98.7	371356		30088
	8046	0.155	0.1075		0.105	0.2	0	0	1	0.275	
	98.725		371054		28739		7957	0.2375		0.11	
	0.0725		0.3	0	0	1.175	2.525	96.3	378520		
	336784		22259		0.2125		0.1325		0.12		
CPU 0.95	0	0		0.55	42.15	57.2	414198		73822		37009
	0.89	0.3225		0.23	0.35	0	0	1.075	2.4	96.625	
	377886		335750		22230		0.18	0.14	0.125	0.1	0
	0	0.875	0.375	98.75	370980		29545		8001	0.215	0.095
	0.07	0	0.075	0	0.95	0.45	98.6	370720		28448	
	7915	0.205	0.115	0.1075		0.375	0.175	0	1.275	2.325	96.4
	377864		335652		22214		0.1975		0.1325		0.1075
CPU 1.525	0.7	0		1.375	43.325		55.225		414134		73443
	37623		0.975	0.385	0.3025		0.2	0	0	1.05	2.45
	96.6	377912		336142		22231		0.24	0.1475		0.135
	0.175	0	0	0.9	0.425	98.675		371180		29649	
	8004	0.1075		0.0575		0.0525		0.125	0	0	1
	0.6	98.4	370626		28641		7916	0.24	0.14	0.125	0.225
	0.325	0	1.25	2.6	96.15	378124		336828		22204	
	0.175	0.1	0.105								

CPU	0.35	0.125	0	0.625	31.825	67.45	414840	74204	
	37445		0.53	0.265	0.2175	0.175	0.1	0	1.1 2.075
	96.925		378272		336562	22251		0.32	0.1525
	0.1275		0.1	0	0	0.875	0.3	98.825	371148
	30063		8053	0.14	0.0725	0.055	0.075	0.125	0 1
	0.125	98.875		370926	28547		7907	0.2725	0.12
	0.0975		0.35	0	0	1.175	2.2	96.625	378044
	336248		22234		0.3125	0.1575		0.1275	
CPU	0.5	0.125	0	1.45	52.5	45.95	414704	75065	38061
	0.955	0.335	0.22	0.2	0	0	1.05	2.3	96.75 378246
	336626		22264		0.21	0.13	0.1225	0.2	0.125 0
	0.925	0.25	98.825		371292	29561		7991	0.1875
	0.0725		0.0525		0.1	0.1	0	1.025	0.35 98.625
	371240		28361		7944	0.17	0.09	0.08	0.15 0 0
	1.25	2.425	96.325		378316	336512		22242	0.38
	0.175	0.1425							
CPU	0.55	0.075	0	0.2	43.675	56.025		414588	73695
	37002		0.885	0.4175		0.32	0.1	0	0 1.1 2.425
	96.575		378100		336278	22236		0.1825	
	0.1075		0.0925		0	0	0	0.85	0.25 98.9
	371200		29942		8002	0.1725		0.075	0.06 0.25 0
	0	0.975	0.275	98.75	371022	28749		7908	0.165 0.08
	0.06	0.075	0	0	1.225	2.55	96.225		378096
	336304		22211		0.2325	0.1375		0.1175	
CPU	1.525	0.1	0	1.35	23.15	75.425		414056	74382
	36907		0.9675		0.4375	0.37	0.225	0.225	0 1.075
	2.55	96.475		378000	336274	22227		0.315	
	0.1275		0.1025		0.3	0	0	0.9	1.175 97.925
	370880		29751		8012	0.22	0.0975		0.0775 0.225
	0.175	0	1	0.5	98.5	370722	28546		7918 0.205
	0.0825		0.06	0.1	0.25	0	1.3	2.45	96.25 377896
	336212		22226		0.26	0.125	0.115		
CPU	1.75	0.125	0	0.575	26.2	73.125		414460	75894
	38093		1.125	0.395	0.29	0.075	0.2	0	1.075 2.475 96.55
	378236		336596		22258	0.2425		0.1275	
	0.1025		0	0.075	0	0.925	0.65	98.425	371092
	29750		7997	0.175	0.0875		0.075	0.525	0.1 0 0.95
	0.65	98.4	370910		28361	7895	0.2725		0.1075

	0.07	0.075	0.225	0	1.15	2.5	96.35	378208	336804	
	22221		0.1975		0.1125		0.1			
CPU	1.2	0.3	0	0.675	22.175		77.05	414434	74005	
	37767		0.85	0.2975		0.215	0.325	0.1	0	1.15 2.5
	96.45	378018		336034		22213		0.3475	0.1375	
	0.1075		0.125	0	0	0.925	0.575	98.5	371496	29754
	8002	0.185	0.085	0.07	0.175	0.325	0	1.025	0.425	98.55
	370952		28735		7932	0.32	0.105	0.075	0.175	0 0
	1.275	2.475	96.25	377768		335476		22194		0.2575
	0.1325		0.1175							

DISQUE	0.725	4.875	0.75	2.575	2.45	0.75	97	94	19	0
92	93									
DISQUE	1	0.95	0.15	1.725	3.475	1.05	97	94	19	0
92	93									
DISQUE	0.825	5.15	0.775	2.175	3.35	1	97	94	19.75	0
92	93									
DISQUE	0.7	9.275	1.325	1.85	5.475	1.725	97	94	20	0
92	93									
DISQUE	1.325	5	0.75	1.7	3.55	1.025	97	94	18.5	0
92	93									
DISQUE	0.75	4.25	0.675	2.05	1.975	0.55	97	94	18	0
92	93									
DISQUE	0.7	3.275	0.475	1.2	3.725	1.125	97	94	18	0
92	93									
DISQUE	0.7	9.25	1.425	1.325	3.575	1.05	97	94	18	0
92	93									
DISQUE	1.15	1.875	0.275	0.75	6.125	1.875	97	94	18	0
92	93									
DISQUE	0.8	9.175	1.45	2.325	4.325	1.35	97	94	18	0
92	93									
DISQUE	0.375	1.3	0.2	1.175	3.375	1.05	97	94	18	0
92	93									
DISQUE	0.65	8.65	1.325	0.875	5.2	1.65	97	94	18	0
92	93									
DISQUE	1.275	5.025	0.775	1.775	5.5	1.725	97	94	18	0
92	93									

DISQUE	1.025	4.575	0.675	1.45	2.8	0.9	97	94	18	0
92	93									
DISQUE	0.65	4.575	0.675	1.275	4.2	1.275	97	94	18.75	0
92	93									
DISQUE	0.75	8.275	1.25	2.475	3.55	1.075	97	94	19	0
92	93									
DISQUE	0.625	5.625	0.85	0.425	5.5	1.7	97	94	19	0
92	93									
DISQUE	1.05	4.775	0.725	1.175	2.95	0.9	97	94	19	0
92	93									
DISQUE	0.65	5.15	0.75	0.875	6.4	2	97	94	19	0
92	93									
DISQUE	0.7	7.825	1.175	2.55	1.725	0.55	97	94	19	0
92	93									
DISQUE	0.775	4.2	0.65	2.25	1.35	0.45	97	94	19	0
92	93									
DISQUE	1.2	4.85	0.75	1.95	3.9	1.225	97	94	19	0
92	93									

MEMOIRE	2413.6	16000	5603.4	4724.2	1661.7
	1349.68	1774.75	5325	6216.5	4527.75
MEMOIRE	2534.65	16000	5724.55	4728.15	1864.85
	1342.25	1897.6	5319.73	6344.4	4527.58
MEMOIRE	2704.35	16000	5839.8	4735.2	1921.4
	1338.58	2025.6	5321.25	6396.9	4523.33
MEMOIRE	2620.6	16000	5832.85	4714.02	1974.1
	1343.83	2023.7	5314.6	6480.35	4495.03
MEMOIRE	2614	16000	5783.1	4728.45	2000.4
	1972.75	5324.3	6463.3	4518.13	1335
MEMOIRE	2670	16000	5805.65	4733.38	2017.75
	1336.5	2038.5	5311.9	6487.65	4520.77
MEMOIRE	2651.25	16000	5849.6	4723.95	2005
	1350.45	2052.85	5315.88	6455.15	4525.78
MEMOIRE	2631.55	16000	5774.45	4728.78	2048.25
	1330.03	2001.15	5338.52	6399.7	4523.58
MEMOIRE	2520.8	16000	5670.5	4720.65	1949.7
	1328.68	1896.65	5316.13	6390	4508.67

MEMOIRE	2659.9	16000	5758.55	4733.08	2001.2
	1334.2	2015.85	5305.08	6496.5	4514.03
MEMOIRE	2477.05	16000	5665.15	4728.58	1914.45
	1332.48	1918.3	5320.55	6325.75	4515.35
MEMOIRE	2599.7	16000	5769.55	4722.13	2044.35
	1325.25	2041.45	5329.33	6449.8	4506.53
MEMOIRE	2669.5	16000	5801	4714.95	2048 1329.63
	2067.15	5311.27	6432.6	4511.63	
MEMOIRE	2642.55	16000	5840.6	4644.4	1977.6
	1337.38	2060.75	5314.95	6431.85	4503.27
MEMOIRE	2688.6	16000	5827	4665.63	2049.4
	1339.75	2093.7	5334.55	6407.55	4526.78
MEMOIRE	2536.1	16000	5744.1	4660.08	2018.5
	1336.85	2004.1	5336.25	6358.45	4511.52
MEMOIRE	2624.7	16000	5760.1	4664.45	1963.05
	1337.23	2032.25	5340.38	6414.25	4525.18
MEMOIRE	2423.3	16000	5574.6	4661.7	1907.5
	1353.75	1911.75	5325.58	6268.35	4509.77
MEMOIRE	2609.15	16000	5777.55	4658.05	2075.95
	1349.05	2123.65	5339.98	6404.35	4518.98
MEMOIRE	2392.65	16000	5604.9	4657.05	1938.3
	1344.4	1926.35	5342.2	6254.85	4523.93
MEMOIRE	2683.65	16000	5804.5	4663.23	2112.4
	1340.7	2236.85	5311.95	6412.6	4516.88
MEMOIRE	2636.85	16000	5804.4	4659.9	2078.55
	1339.35	2133.15	5337.4	6421.6	4515

PAGE	0	392	0	0	79770.3	37111.3	0	294
	0	0	15093.3	23163.3	0	280	0	0
	13306.3	25722.4	0	280	0	0	14654.3	
	24305.5	0	294	0	0	4015	3579.18	
PAGE	0	392	0	0	78579.5	36378.5	0	280
	0	0	14814	23060.7	0	280	0	0
	13282.3	25596	0	280	0	0	14386.5	
	24191.1	0	280	0	0	3995	3576.73	
PAGE	0	392	0	0	77424	35674.4	0	280
	0	0	14545	22959	0	280	0	0

	13258.8	25470.7	0	266	0	0	14134	
	24080.6	0	280	0	0	3974.75	3574.25	
PAGE	0	378	0	0	76312	35002.8	0	266
	0	0	14287.5	22858.3	0	266	0	0
	25345.2	0	294	0	0	13885.8	23968.4	0
	280	0	0	3955.5	3571.8			
PAGE	0	392	0	0	75241	34361.8	0	294
	0	0	14036.3	22758.3	0	280	0	0
	13211.5	25222.4	0	280	0	0	13650.8	
	23858.6	0	280	0	0	3936	3569.4	
PAGE	0	392	0	0	74199.3	33743.8	0	280
	0	0	13797	22659.3	0	294	0	0
	25100.8	0	280	0	0	13424.5	23751.1	0
	280	0	0	3916.75	3566.9			
PAGE	0	406	0	0	73177.3	33142.3	0	280
	0	0	13562.5	22562.3	0	280	0	0
	13164.8	24980.4	0	280	0	0	13203.5	
	23643.3	0	280	0	0	3897.5	3564.5	
PAGE	0	378	0	0	72174.3	32556.8	0	280
	0	0	13335.3	22463.7	0	280	0	0
	13141.8	24862.6	0	280	0	0	12986.3	23535
	0	280	0	0	3878.75	3562		
PAGE	0	392	0	0	71206.8	31996.6	0	280
	0	0	13117	22367.2	0	280	0	0
	13118.5	24744.4	0	266	0	0	12778.3	
	23429.1	0	280	0	0	3860.25	3559.63	
PAGE	0	406	0	0	70265.3	31455.4	0	266
	0	0	12906.8	22272.8	0	280	0	0
	13095.5	24627.4	0	294	0	0	12577.8	
	23324.2	0	280	0	0	3841.75	3557.2	
PAGE	0	378	0	0	69356	30936.7	0	294
	0	0	12704.8	22177.9	0	280	0	0
	13072.3	24510	0	280	0	0	12388.8	
	23222.7	0	266	0	0	3823	3554.7	
PAGE	0	406	0	0	68454.5	30425.9	0	280
	0	0	12506.3	22083.8	0	280	0	0
	13049.5	24396.6	0	280	0	0	12194.5	
	23117.1	0	280	0	0	3805.5	3552.35	

PAGE	0	378	0	0	67583.8	29936	4	348
	0	0	12312.5	21974.4	0	266	0	0
	13026.8	24282.8	0	280	0	0	12014.8	
	23014.8	0	294	0	0	3787.5	3549.93	
PAGE	0	406	0	0	66749.5	29470.1	0	280
	0	0	12123.5	21832.6	0	294	0	0
	13003.5	24168.7	0	280	0	0	11836.3	
	22914.8	0	280	0	0	3769.25	3547.53	
PAGE	0	392	0	0	65906.8	29002.2	0	268
	0	0	11941.5	21740.3	0	280	0	0
	12981.3	24059.8	0	280	0	0	11661.8	
	22813.2	0	266	0	0	3751.5	3545.05	
PAGE	0	378	0	0	65106.5	28560.7	0	280
	0	0	11768.8	21646.6	0	266	0	0
	12958.3	23946.4	0	280	0	0	11494.8	
	22713.6	0	294	0	0	3734.25	3542.6	
PAGE	0	392	0	0	64318.3	28128.7	0	294
	0	0	11596	21556.9	0	294	0	0
	12935.8	23836.7	0	280	0	0	11334.3	22615
	0	280	0	0	3716.75	3540.23		
PAGE	0	406	0	0	63542.3	27705.8	0	280
	0	0	11430.8	21468	0	280	0	0
	12913.3	23728.1	0	280	0	0	11176.5	
	22518.4	0	280	0	0	3699.75	3537.8	
PAGE	0	392	0	0	62791	27299.1	0	280
	0	0	11271.5	21379.9	0	280	0	0
	12890.8	23620.5	0	280	0	0	11022	
	22420.2	0	280	0	0	3682.75	3535.4	
PAGE	0	392	0	0	62063.8	26907.6	0	280
	0	0	11116	21293.5	0	280	0	0
	12868.8	23515.1	0	280	0	0	10875.5	
	22326.5	0	280	0	0	3665.75	3533	
PAGE	0	392	0	0	61347.5	26523.8	0	280
	0	0	10963.8	21206.8	0	280	0	0
	12846.5	23408.1	0	280	0	0	10725	
	22228.8	0	280	0	0	3649	3530.6	
PAGE	0	392	0	0	60641	26147.5	0	280
	0	0	10813	21119.7	0	280	0	0
								12824

23303.3	0	280	0	0	10585	22134.3	0
266	0	0	3632.5	3528.2			

RESEAU	20177	0	20168	0	2	5590	0	5456
0	24	5675	0	5540	0	12	5539	0
13	5595	0	5457	0	23			
RESEAU	20150	0	20140	0	4	5608	0	5466
0	27	5700	0	5557	0	14	5567	0
14	5625	0	5485	0	29			
RESEAU	20468	0	20633	0	3	5610	0	5473
0	22	5689	0	5550	0	7	6000	0
17	5614	0	5472	0	36			
RESEAU	21010	0	21654	0	4	6131	0	5748
0	23	6130	0	5800	0	15	5554	0
16	6121	0	5743	0	28			
RESEAU	20187	0	20172	0	0	5609	0	5468
0	18	5682	0	5539	0	12	5548	0
22	5605	0	5468	0	39			
RESEAU	20158	0	20141	0	1	5612	0	5465
0	29	5687	0	5548	0	23	5550	0
14	5613	0	5472	0	25			
RESEAU	20219	0	20205	0	6	5602	0	5460
0	14	5688	0	5547	0	20	5551	0
27	5603	0	5469	0	25			
RESEAU	20213	0	20189	0	1	5606	0	5467
0	39	5678	0	5540	0	11	5544	0
21	5607	0	5470	0	16			
RESEAU	20195	0	20182	0	4	5591	0	5458
0	15	5668	0	5533	0	18	5541	0
16	5591	0	5452	0	29			
RESEAU	20158	0	20149	0	0	5618	0	5480
0	28	5681	0	5547	0	14	5555	0
16	5613	0	5477	0	29			
RESEAU	20168	0	20154	0	3	5596	0	5460
0	20	5684	0	5547	0	10	5544	0
12	5600	0	5463	0	34			

RESEAU	20195	0	20178	0	2	5618	0	5483
0	30	5696	0	5554	0	17	5567	0
12	5611	0	5473	0	23			
RESEAU	20224	0	20239	0	1	5648	0	5489
0	27	5679	0	5542	0	12	5546	0
15	5599	0	5464	0	15			
RESEAU	20171	0	20162	0	1	5621	0	5484
0	27	5691	0	5549	0	11	5556	0
14	5610	0	5472	0	35			
RESEAU	20194	0	20184	0	0	5617	0	5478
0	21	5685	0	5547	0	13	5545	0
12	5607	0	5468	0	21			
RESEAU	20183	0	20172	0	3	5621	0	5478
0	31	5680	0	5541	0	13	5547	0
17	5613	0	5467	0	23			
RESEAU	20266	0	20245	0	0	5618	0	5479
0	22	5678	0	5543	0	13	5548	0
18	5614	0	5469	0	31			
RESEAU	20190	0	20169	0	1	5633	0	5491
0	22	5701	0	5558	0	11	5566	0
28	5622	0	5483	0	37			
RESEAU	20194	0	20186	0	1	5611	0	5473
0	23	5680	0	5538	0	14	5533	0
25	5603	0	5464	0	26			
RESEAU	20144	0	20135	0	1	5611	0	5474
0	22	5674	0	5535	0	12	5545	0
13	5607	0	5470	0	24			
RESEAU	20195	0	20179	0	0	5612	0	5475
0	20	5684	0	5542	0	11	5551	0
17	5610	0	5474	0	22			
RESEAU	20228	0	20208	0	1	5621	0	5478
0	24	5680	0	5543	0	7	5553	0
21	5611	0	5468	0	26			

H.8. FICHER AUX3.DAT

Tue May 1 00:59:03 WET DST 1990

CPU	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0					
CPU	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0					
CPU	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0					
CPU	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0					
CPU	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0					
CPU	1.13646	0.1375	0	14.4552	23.0198	7.37604				
	48.2208	265.119	47.8208	0.424583	0.210729					
	0.181875	0.0135417	0.0197917	0	0.0927083	0.1125				
	3.96563	1.14479	9.37083	0.902083	0.0307292					
	0.0192708	0.0177083	0	0	0	0	0	0	0	0
	0	0	0	0	0.0270833	0	0	0.158333		

	0.259375	3.74896	1.40104	10.3	0.953125	0.015625				
	0.0127083	0.0134375	0.00833333	0	0	0.046875				
	0.117708	4.00625	0.803125	6.75417	0.5375					
	0.0128125	0.00520833	0.0040625							
CPU	2.65417	0.267708	0	36.349	53.6302	10.1437				
	108.22	652.133	112.181	0.850625	0.335208					
	0.256146	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0					
CPU	2.78021	0.272917	0	36.0302	53.9406	10.1521				
	108.916	657.525	112.356	0.910729	0.366979					
	0.273854	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0					
CPU	2.60938	0.382292	0	33.0646	48.6969	14.1969				
	108.534	601.397	114.447	0.857396	0.402396					
	0.326979	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0					
CPU	2.32396	0.342708	0	32.9104	49.1219	18.0625				
	112.073	609.947	119.149	0.79375	0.395417					
	0.326146	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0					
CPU	2.61354	0.309375	0	32.9281	49.3885	17.7635				
	110.488	607.987	115.831	0.797917	0.425521					
	0.366979	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0

	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0					
CPU	2.50625	0.304167	0	33.0552	48.8719	18.1635					
	109.968	615.504	114.626	0.765417	0.396667						
	0.337604	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0						
CPU	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0							
CPU	1.55312	0.248958	0	22.2802	33.749	12.7896					
	75.2677	405.458	79.2437	0.509167	0.275208						
	0.242083	373.416	1781.26	0	0	0	0	0			
	0	0	0	929.13	0.0364583	1023.38					
	1.74375	504.365	615.601	21.5438	50.9531						
	13.1437	0.174375	7.16906	2.38219	968.65						
	1583.36	652.01	1.225 4.60833	4.83333	17.8833						
	47.4417	12.625	0.156667	6.36333	2.10917						
	12392.8	1387.14	11684.8	1.10833	557.95						
	535.31	16.4573	38.2885	11.3167	0.143646						
	5.58031	1.85427									
CPU	1.55312	0.248958	0	22.2802	33.749	12.7896					
	75.2677	405.458	79.2437	0.509167	0.275208						
	0.242083	373.416	1781.26	0	0	0	0	0			
	0	0	0	929.13	0.0364583	1023.38					
	1.74375	504.365	615.601	21.5438	50.9531						
	13.1437	0.174375	7.16906	2.38219	968.65						
	1583.36	652.01	1.225 4.60833	4.83333	17.8833						
	47.4417	12.625	0.156667	6.36333	2.10917						
	12392.8	1387.14	11684.8	1.10833	557.95						
	535.31	16.4573	38.2885	11.3167	0.143646						
	5.58031	1.85427									

CPU	1.55312	0.248958	0	22.2802	33.749	12.7896
	75.2677	405.458	79.2437	0.509167	0.275208	
	0.242083	373.416	1781.26	0	0	0
	0	0	0	929.13	0.0364583	1023.38
	1.74375	504.365	615.601	21.5438	50.9531	
	13.1437	0.174375	7.16906	2.38219	968.65	
	1583.36	652.01	1.225	4.60833	4.83333	17.8833
	47.4417	12.625	0.156667	6.36333	2.10917	
	12392.8	1387.14	11684.8	1.10833	557.95	
	535.31	16.4573	38.2885	11.3167	0.143646	
	5.58031	1.85427				

CPU	1.55312	0.248958	0	22.2802	33.749	12.7896
	75.2677	405.458	79.2437	0.509167	0.275208	
	0.242083	373.416	1781.26	0	0	0
	0	0	0	929.13	0.0364583	1023.38
	1.74375	504.365	615.601	21.5438	50.9531	
	13.1437	0.174375	7.16906	2.38219	968.65	
	1583.36	652.01	1.225	4.60833	4.83333	17.8833
	47.4417	12.625	0.156667	6.36333	2.10917	
	12392.8	1387.14	11684.8	1.10833	557.95	
	535.31	16.4573	38.2885	11.3167	0.143646	
	5.58031	1.85427				

CPU	2.36458	0.604167	0	48.9677	12.2333	21.7708
	633880	279.294	60.8562	1.87427	1.23219	
	1.09323	0.229167	0.195833	0.0114583	1.81771	
	3.18229	95.0052	23.1958	149.544	15.099	
	0.313437	0.208333	0.188958	0.296875	0.107292	0
	3.0125	4.91458	91.9875	25.0219	179.149	
	15.3521	0.26875	0.150208	0.128021	0.291667	
	0.151042	0	3.59479	5.62604	90.801	30.6396
	174.598	18.3969	0.323542	0.185	0.157917	0.26875
	0.161458	0	3.42187	6.64479	90.0479	31.8833
	189.424	20.2406	0.322813	0.204688	0.181979	

CPU	2.8875	0.838542	0	61.1729	1.11771	9.10104
	1.06634e+06	114.092	30.7292	2.74031	2.02448	
	1.87396	0.258333	0.227083	0.00520833	2.36458	
	3.50417	94.1313	21.3656	146.014	14.6302	
	0.338438	0.217708	0.194896	0.201042	0.21875	

	0.00208333	3.175	4.37708	92.349	22.0854	166.058
	15.099	0.340104	0.1925	0.16375	0.315625	
	0.202083	0	2.55417	5.14687	92.3635	23.4167
	160.945	16.1167	0.401771	0.231979	0.204896	
	0.240625	0.204167	0	4.04167	6.11875	89.8969
	31.8031	175.875	19.4542	0.433542	0.290938	0.265
CPU	3.47812	0.861458	0	61.9302	0.871875	8.91771
	1.01628e+06	113.097	29.7625	3.57635	2.83156	
	2.67438	0.263542	0.382292	0.00729167	3.89896	
	4.08229	92.0271	27.5896	154.049	17.7719	
	0.406042	0.277083	0.250833	0.202083	0.186458	
	0.00208333	2.81875	4.17396	92.9469	19.4208	
	172.149	13.6781	0.311667	0.189688	0.161563	
	0.285417	0.198958	0.0229167	2.66771	4.90729	
	92.5302	25.101	148.469	16.426	0.399687	
	0.265938	0.231771	0.259375	0.197917	0	3.34479
	5.48021	91.1865	25.0135	163.708	17.1229	
	0.387396	0.255938	0.233542			
CPU	2.925	0.248958	0	60.6979	0.896875	8.38021
	577329	90.1417	27.124	2.79396	2.34125	
	2.25354	98.7906	156.337	57.7708	2.39271	
	29.7417	131.75	5235.74	126.496	11.251	
	0.218958	0.142604	0.127188	250.751	162.526	
	0.00208333	1.41458	2.47396	87.7781	13.3469	
	159.679	11.3563	0.228125	0.139375	0.122292	
	309.389	136.739	0	1.14167	2.18438	88.4323
	14.7146	124.003	10.775	0.215104	0.132292	
	0.119375	458.473	57.0779	0	0.98125	2.4125
	88.2729	15.0448	125.465	11.026	0.203333	
	0.129063	0.118437				
CPU	2.99479	0.202083	0	60.6292	1.00313	8.20625
	579591	94.174	27.2823	2.795	2.33333	2.24552
	0.144792	0.180208	0	1.91563	1.42188	92.4958
	15.4583	130.749	11.3417	0.21375	0.137708	
	0.124063	0.130208	0.0604167	0	1.4875	2.17292
	92.1729	13.2542	159.621	11.5458	0.232396	
	0.144688	0.125833	0.2	0.0791667	0	1.20313
	2.15729	92.5688	15.8375	129.968	11.7156	

	0.214271	0.132083	0.118437	0.194792	0.0458333	0
	1.06563	2.32188	92.4458	15.6896	130.289	
	11.3594	0.204479	0.121667	0.108229		
CPU	0.0541667	0.0208333	0	0.0604167	0.9125	3.18958
	4.125	8.71146	3.12396	0.0317708	0.0138542	0.0104167
	0	0	0	0.0625	0.0729167	4.02708
	5.625	0.540625	0.0119792	0.00645833	0.005625	0.00625
	0.003125	0	0.065625	0.0791667	4.02187	0.697917
	6.39687	0.554167	0.00916667	0.00520833	0.00416667	
	0.0208333	0.00416667	0	0.0447917	0.108333	4.01771
	0.835417	5.63125	0.572917	0.00864583	0.0053125	
	0.00479167	0.0114583	0.00520833	0	0.0760417	0.071875
	4.01875	0.8	5.7375	0.58125	0.00927083	0.0046875
	0.0040625					
CPU	1.62604	1.06042	0	6.86146	25.024	68.0406
	129.478	232.611	109.82	1.54729	0.940833	
	0.831354	0.278125	0.170833	0	2.1375	2.93958
	94.9229	21.1052	150.204	15.4375	0.334167	
	0.236563	0.225	0.284375	0.253125	0	4.475
	90.9677	5884.95	173.4	18.3552	0.414583	0.2675
	0.244583	0.363542	0.288542	0	3.95104	4.79375
	91.3208	31.3646	161.396	19.8438	0.439792	
	0.30625	0.285521	0.611458	0.319792	0.0104167	
	8.84479	17.7531	73.0271	39.9813	209.745	
	158.962	0.754167	0.575938	0.534688		
CPU	1.28229	0.6375	0	6.38854	18.3865	58.5125
	106.227	193.904	103.95	1.27844	0.824583	
	0.720833	200.415	84.1867	0	1.21042	2.20625
	75.7375	15.6104	111.878	10.999	0.21875	
	0.138438	0.126667	270.07	70.5865	0	2.46667
	1.92083	74.5729	6638.8	71.6937	11.0688	
	0.284583	0.155	0.12875	179.618	147.096	0.015625
	1.45729	2.5125	75.1969	17.2437	111.878	
	11.4031	0.289583	0.195208	0.161667	0.670833	
	0.242708	0	12.651	24.6469	41.9063	45.0313
	217.619	38.025	0.801875	0.664479	0.635938	
CPU	2.88646	0.80625	20.1417	6.64375	27.1854	
	65.6365	6.15609e+06	35651148	982011	19.3239	

	0.899583	0.777813	0.303125	0.28125	0.015625	
	3.24271	3.79583	92.9823	176524	214714	16676
	0.417604	0.277813	0.256771	0.265625	0.182292	0
	1.17917	1.08229	97.7396	8144756	8256824	
	534462	0.274063	0.143958	0.1225	0.2	0.144792
	0.0104167	1.77604	3.13333	95.0979	169646	
	112227	18184.1	0.336042	0.211979	0.187813	
	0.803125	0.2625	0.00520833	13.249	27.8625	
	58.9427	478898	904594	166299	0.993333	
	0.836458	0.803958				
CPU	2.49375	0.632292	0	18.5635	15.3323	61.1406
	657237	1.98158e+06		509193	1.98156	1.35302
	1.23875	0.253125	0.160417	0	1.31875	2.69167
	95.9958	379382	337528	23059.6	0.26	0.1425
	0.123854	2.56042	0.126042	1.08469e+06		1.86354
	2.04063	94.9625	379138	93218.7	28603.4	
	0.267708	0.145938	0.121875	0.135417	0.14375	0
	1.22917	0.742708	98.025	375821	53980.4	14160
	0.254167	0.122604	0.101875	1.28125	0.398958	
	0.00625	23.6271	51.626	24.8615	618445	
	1.38905e+06	267145	1.66115	1.42323	1.38417	
CPU	2.56354	0.363542	0	3.45312	19.6708	76.9063
	647112	78320	484155	1.685	1.17698	1.08906
	0.223958	0.0708333	0	1.17188	2.75417	96.0198
	380421	341267	23628.1	0.246354	0.134896	
	0.119687	0.198958	0.122917	0	1.06563	0.667708
	98.2719	373223	34177.9	9274.17	0.224792	
	0.112187	0.0911458	0.116667	0.0979167	0	1.01354
	0.561458	98.4417	371559	28938	8435.08	
	0.227188	0.105417	0.0857292	1.2625	0.29375	0
	22.7854	52.1521	25.1708	606118	1362577	
	257550	1.47594	1.27323	1.23875		
CPU	2.87188	0.202083	0	2.80417	17.2094	75.8448
	619093	73877.3	463123	1.59167	1.10813	
	1.02229	0.241667	0.0885417	0	1.45208	2.4875
	91.7896	364191	325908	22390.8	0.250313	
	0.128125	0.110104	0.133333	0.0604167	0	0.952083
	0.371875	94.5094	355929	27598	7993.42	

	0.178021	0.0884375	0.0730208	0.155208	0.0947917	0				
	0.955208	0.332292	94.5458	355944	27315.4					
	7988.33	0.213958	0.0963542	0.0765625	1.23125					
	0.301042	0	21.9781	50.7396	23.2469	580904				
	1.30742e+06	246854	1.40885	1.22385	1.19313					
CPU	0.0802083	0.0208333	0	0.158333	0.955208	3.05417				
	666813	14234.7	494321	0.0472917	0.0389583					
	0.0388542	0.0135417	0.00520833	0	0.053125	0.096875				
	4.01667	404746	366624	28750.7	0.0108333					
	0.0053125	0.00458333	0.00833333	0	0	0.0385417				
	0.0229167	4.10521	383238	9136.54	6659.79					
	0.0071875	0.003125	0.0021875	0.00416667	0.00208333	0				
	0.0427083	0.0208333	4.10313	125334	23532.5					
	6151.79	0.0122917	0.00572917	0.0040625	0.0510417					
	0.0291667	0	0.955208	2.19062	1.02396	636133				
	1.42818e+06	271997	0.0666667	0.0557292	0.0538542					
DISQUE	0	0	0	0	0	0	0	0	0	0
0	0									
DISQUE	0	0	0	0	0	0	0	0	0	0
0	0									
DISQUE	0	0	0	0	0	0	0	0	0	0
0	0									
DISQUE	0	0	0	0	0	0	0	0	0	0
0	0									
DISQUE	0	0	0	0	0	0	0	0	0	0
0	0									
DISQUE	0.7625	3.28021	0.525	1.26042	4.93438					
	1.14583	41.6563	40.7604	14.0625	0.447917					
	29.7604	10.8021								
DISQUE	1.39479	6.36146	1.03958	1.725	6.75625					
	1.75729	93	91	30.6458	1	68	24.9375			
DISQUE	1.37292	7.26042	1.16562	1.65938	7.10938					
	1.85	93	91	30.5104	1	68	25			
DISQUE	1.51667	8.61458	1.35625	3.31979	11.3969					
	2.98333	89.75	87.2083	29.3542	0.958333	68.1563				
	23.9375									

DISQUE	1.64688		8.47604	1.38021	3.30625	12.325					
	3.20313	94	91	31.3958	1	73.0625	25				
DISQUE	1.51042		8.98229	1.46042	3.1875	9.14479					
	2.46146	94	91	31.9479	1.01042	74.3646	25				
DISQUE	1.37917		7.09792	1.15521	2.73958	7.6					
	2.26667	94	91	30.7708	1	76.1146	25				
DISQUE	0	0	0	0	0	0	0	0	0	0	0
	0	0									
DISQUE	0.361458		0.219792	0.034375	8.02709	12.2948					
	4.32708	27.3719	144.403	27.6396	0.16625						
	3.27208	1.15427									
DISQUE	0.35	0	0	8.3	12.375	4.375	24.475				
	146.725	27.45	0.13	0.11	0.1175						
DISQUE	0.35	0	0	8.3	12.375	4.375	24.475				
	146.725	27.45	0.13	0.11	0.1175						
DISQUE	0.35	0	0	8.3	12.375	4.375	24.475				
	146.725	27.45	0.13	0.11	0.1175						
DISQUE	1.61042		7.35521	1.18646	5.57813	15.3375					
	4.20208	94	91	30.8542	1	75	26.25				
DISQUE	2.00729		10.899	1.76042	9.03333	21.5448					
	5.68125	94	91	30.5521	1	73.3646	25.7396				
DISQUE	2.67604		14.4885	2.38125	8.14896	23.824					
	6.07917	94	91	31.0938	1	77.4271	48.7917				
DISQUE	1.19167		5.82604	0.913542	2.37708	7.27188					
	2.13333	90.0833	87.2083	29.1146	0.958333						
	79.5417	68.0417									
DISQUE	0.961458		5.79167	0.90625	1.60521	7.07188					
	2.03646	90.0833	87.2083	28.9167	0.958333						
	79.5417	68.0417									
DISQUE	0.0552083		0.484375	0.0760417	0.228125	0.45					
	0.132292	4.04167	3.79167	1.41667	0	3.58333					
	2.95833										
DISQUE	2.47292		16.5031	2.73125	9.21042	24.9417					
	6.64479	97	94.9063	34.5417	0	90.0417					
	66.4375										
DISQUE	82.3487		30.0333	2.63437	5.36458	13.8167					
	3.93854	79.8229	82.2917	26.9688	0.0520833						
	75.2604	48.5729									

DISQUE	2.40104	13.4562	2.24167	7.58021	18.9698					
	5.17708	97	97.625	31.3958	0	92.0313				
	63.5521									
DISQUE	1.67604	9.55417	1.55521	5.61354	11.2177					
	3.25	97	94	30.6042	0	92	64.6354			
DISQUE	1.20521	7.28542	1.16875	2.425	8.54167					
	2.61563	97	94	30.5	0	92	65			
DISQUE	1.03542	5.98125	0.934375	1.45313	6.50208					
	1.97917	92.9583	90.0833	29.1458	0	88.1667				
	62.2917									
DISQUE	0.0552083	0.419792	0.06875	0.226042	0.226042					
	0.0697917	4.04167	3.91667	1.29167	0	3.83333				
	2.70833									
MEMOIRE	0	0	0	0	0	0	0	0	0	0
MEMOIRE	0	0	0	0	0	0	0	0	0	0
MEMOIRE	0	0	0	0	0	0	0	0	0	0
MEMOIRE	0	0	0	0	0	0	0	0	0	0
MEMOIRE	0	0	0	0	0	0	0	0	0	0
MEMOIRE	1476.52	6590.97	395.692	70.0187	0	0				
	468.583	52.4813	6915.6	171.156						
MEMOIRE	3188.49	15224	0	0	0	0	0	0	0	0
	0	0								
MEMOIRE	3224.89	15217.7	0	0	0	0	0	0	0	0
	0	0								
MEMOIRE	3502.65	17233.1	0	0	0	0	0	0	0	0
	0	0								
MEMOIRE	3172.97	19875	0	0	0	0	0	0	0	0
	0	0								
MEMOIRE	3131.31	19338.5	0	0	0	0	0	0	0	0
	0	0								
MEMOIRE	3087.75	19190.6	0	0	0	0	0	0	0	0
	0	0								
MEMOIRE	2081.08	13219.8	116612	0.178125	1112.93					
	1783.78	453.015	1585.49	21086.8	1187.53					
MEMOIRE	1.21354	4.50625	0.684375	2.6375	8.05208					
	2.28229	61.6875	59.7188	483	0.65625					

MEMOIRE	0.361458	0.219792	0.034375	8.02709	12.2948
	4.32708	27.3719	144.403	663.35	0.16625
MEMOIRE	0.35	0	0	8.3	12.375
	146.725	658.8	0.13	4.375	24.475
MEMOIRE	0.35	0	0	8.3	12.375
	146.725	658.8	0.13	4.375	24.475
MEMOIRE	3833.05	14651	5610.1	3772.79	7588.99
	1330.31	7711.28	3494.63	172887	3275.65
MEMOIRE	4071.73	14457.3	6190.13	3621.81	7687.6
	1730.05	8108.39	3143.02	190265	3070.53
MEMOIRE	4076.28	14354.2	5743.56	3310.93	9007.94
	1524.11	7706.68	3213.29	182624	3368.22
MEMOIRE	3553.29	14351	5671.62	3562.27	10092.7
	1489.91	6886.58	3415.78	144174	3654.13
MEMOIRE	3654.24	14339.6	5889.9	3700.15	10541.7
	1670.36	7253.2	3541.73	152222	3785.59
MEMOIRE	139.454	583.333	209.56	161.248	500
	76.0013	329.885	156.965	6731.9	169.604
MEMOIRE	3654.73	14175	5046.06	3018.67	7406.27
	3029.22	9148.34	2812.27	175157	3373.19
MEMOIRE	3437.51	11651.7	4482.38	3426.95	3520.9
	2747.63	5857.65	3098.98	127855	2866.98
MEMOIRE	3663.59	14536.5	5923.47	3777.32	3096.37
	3779.57	5591.93	4223.8	156306	3844.41
MEMOIRE	3496.26	14779.2	6047.79	4531.58	110893
	3260.1	1956.65	6352.87	162145	3875.57
MEMOIRE	3272.01	15000	5924.96	4440.58	1751.39
	4336.29	2031.08	7749.34	142168	4526.21
MEMOIRE	3189.83	14375	5646.88	4331.61	1618.81
	4185.13	2047.71	7426.63	135569	4336.54
MEMOIRE	140.058	625	245.419	187.174	77.7188
	180.803	78.9437	211.478	6039.3	185.885
PAGE	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0

PAGE	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0								
PAGE	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0								
PAGE	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0								
PAGE	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0								
PAGE	0	0.00416667	0	0	3794.23	4973.97	0			
0.003125	0	0	332	180.918	0	0	0	0		
0	0	0.0166667	0.003125	0.4	0.290625	144.542				
79.3292	0.0229167	0.00416667	0	0	134.802					
73.0388										
PAGE	0	0	0	0	8350	10866.2	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
PAGE	0	0	0	0	8350	10866.2	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
PAGE	0.180208	0.372917	0	0	7446.02	9038.5				
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0									
PAGE	0.059375	0.186458	0	0	7173.68	8217.38				
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0									
PAGE	0.0125	0.00625	0	0	6679.77	7820.66				
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0									
PAGE	0.00520833	0.00625	0	0	6248.9	7536.7				
0	0	0	0	0	0	0	0	0	0	

	0	0	0	0	0	0	0	0	0	0
	0	0								
PAGE	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0		
PAGE	265.029	1601.11	0	0.297917	0.595833					
	0.1375	21.5417	20.8542	7.5625	0.229167	16.5				
	5.72917	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0		
PAGE	1.21354	4.50625	0.684375	2.6375	8.05208					
	2.28229	61.6875	59.7188	20.125	0.65625					
	49.4167	16.4063	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
PAGE	0.361458	0.219792	0.034375	8.02709	12.2948					
	4.32708	27.3719	144.403	27.6396	0.16625					
	3.27208	1.15427	954.285	4552.1	0	0	0			
	0	0	0	0	0	0	2374.46	0.0958333		
	2615.31	4.45625	1448.79	1767.67						
PAGE	0.35	0	0	8.3	12.375	4.375	24.475			
	146.725	27.45	0.13	0.11	0.1175	995.775	4750.02			
	0	0	0	0	0	0	0	0	0	0
	2477.7	0.1	2729.02	4.65	1511.78	1844.53				
PAGE	0.283333	0.108333	0	0	2941.15	5085.11				
	0.83125	1.50833	2.13125	2.26875	3642.96					
	3546.24	0.00416667	0.0583333	0.441667	0.490625					
	5820.47	5208.41	0.639583	0.70625	1.17604					
	1.00729	2076.05	1207.55	0.791667	0.85	3.12708				
	1.87188	1589.38	1042.79							
PAGE	0.08125	0.247917	0	0	2814.36	4728.44				
	0.709375	0.65625	3.79583	2.80625	2852.95					
	2563.86	0.210417	0.202083	1.67188	0.764583					
	5476.02	4768.97	0.565625	0.355208	1.60833					
	1.41771	1891.38	1134.68	1.14063	1.60938	4.075				
	4.39271	1413.5	941.604							
PAGE	0.0114583	0.0989583	0	0	2733.73	4404.21				
	1.44687	1.75833	1.61042	2.0875	2369.01					
	2065.9	0.0885417	0.065625	0.25625	0.541667					
	5130.35	4368.26	0.541667	0.695833	1.83542					

	1.42813	1685.9	1040.24	1.15521	0.935417	
	3.23542	2.98854	1258.6	838.474		
PAGE	0.01875	0.127083	0	0	2574.52	3978.06
	99.7813	156.333	60.9604	0.672917	1898.87	
	1651.18	84.9896	0.084375	83.4677	0.197917	
	4567.95	3823.97	63.2802	0.071875	63.4583	0
	1419.69	903.353	86.4136	0.0770833	83.75	0
	1081.81	728.106				
PAGE	0.015625	0.116667	0	0	2542.07	3789.88
	0.380208	0.084375	0	0	1893.17	1714.76
	0.0833334	0	0	4711.18	3926.74	0.43125
	0.0864583	0	0	1431.72	940.998	0.405208
	0.0802084	0	0	1099.46	758.985	
PAGE	0	0.00416667	0	0	90.1667	145.004
	0.0197917	0.0708333	0	0	27192.1	26035.8
	0.00625	0	0	200.594	166.035	0.03125
	0.084375	0	0	58.5625	39.0781	0.0229167
	0.0135417	0	0	44.7813	30.7198	
PAGE	0.226042	0.45	0	0	2137.82	3389.17
	0.68125	0.223958	1.98229	2.49167	84997.4	
	166436	0.194792	0.3125	0.672917	0.380208	
	4434.19	3668.65	1.19167	0.930208	2.575	2.66458
	1326.35	885.679	1.54167	2.90104	3.62396	
	3.52188	19423.7	16701.7			
PAGE	33.3354	0.08125	28.4792	0	1644.73	
	2616.96	84.9594	0.210417	70.2698	0.759375	
	10465.5	8936.59	164.169	0.720833	155.069	
	0.457292	2849.88	2444.65	913.858	1.15938	
	891.302	1.11354	8919.18	8044.44	0.934375	
	3.04583	0.926042	0.977083	17655	16407.1	
PAGE	10.1896	3873.43	365.958	70.3958	1851.8	
	2946.27	99.1031	1511.56	847.443	283.399	
	46885.8	36862.4	550.5	15354.7	6031.3	2153
	3132.47	2745.68	91.8354	2374.06	1264.26	
	355.975	42267.4	38352.5	184.76	3654.03	
	1293.43	485.02	8773.22	6613.41		
PAGE	0	638.083	0	0	1719.69	2792.16
	6.95833	528.667	126.083	20.75	14039.8	12183.2

	14.2083	1322.25	520.917	796.792	51472.1	
	65024.1	4.04167	992.833	180.167	33.0417	
	25966.3	37260.9	92.375	2878.58	863.583	
	343.542	5450.44	3494.21			
PAGE	0	395.5 0	0	1602.31	2676.32	0
	375.083	0 0	9700.12	11212	0	366.75
	0 0	30740.1	71768.4	0 288	0 0	0
	13234.7	26313.4	0.0833333	859.333	0 0	0
	4479.78	2799.29				
PAGE	0	375.667	0 0	1440.23	2480.3	1
	464.25	0 0	7238.24	10053.2	0	268.333
	0 0	15258.7	49557.9	0	268.333	0 0
	9105.61	22473.6	0	820.333	0 0	3842.13
	2430.23					
PAGE	0	102.583	0 0	60.5625	105.653	
	138.042	2073.17	1117	363.75	226.385	276.981
	0 31.6667	0 0	524.563	1895.09	12.875	
	580.5 430.75	124.542	4987.92	3947.15	97.1667	
	2203.08	970.667	472.125	143.156	85.8883	
RESEAU	0	0 0	0 0	0 0	0 0	0
	0 0	0 0	0 0	0 0	0 0	
RESEAU	0	0 0	0 0	0 0	0 0	0
	0 0	0 0	0 0	0 0	0 0	
RESEAU	0	0 0	0 0	0 0	0 0	0
	0 0	0 0	0 0	0 0	0 0	
RESEAU	0	0 0	0 0	0 0	0 0	0
	0 0	0 0	0 0	0 0	0 0	
RESEAU	13743.3	0	12813.3	0	0.375 255.75	0
	246.875	0 0	0 0	0	0 192.875	
	0 186.958	0	0 138.958	0	138.458	0
	0.166667					
RESEAU	24163.3	0	23717.1	0	1.58333	0 0
	0 0	0 0	0 0	0 0	0 0	0
	0 0	0 0	0 0	0		

RESEAU	24106.3	0	23658	0	1.04167	0	0				
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0					
RESEAU	27581.8	0.0416667	27556.4	0	4.79167	0					
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0				
RESEAU	27418.2	0	27896.3	0	3.5	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0						
RESEAU	23709.3	0	24007.8	0	4	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0						
RESEAU	21631.1	0	22143.8	0	2.75	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0						
RESEAU	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0								
RESEAU	1874.29	12029.2	142526	0.217708	1381.23						
2180.17	535.829	1981.88	1317.92	1781.29	0						
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0									
RESEAU	0.4	0.075	0	0.325	0.65	0.15	23.5	22.75	8.25	0.25	
18	6.25	0	0	0	0	0	0	0	0	0	0
0	0	0	0								
RESEAU	1.21354	4.50625	0.684375	2.6375	8.05208						
2.28229	61.6875	59.7188	20.125	0.65625							
49.4167	16.4063	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0						
RESEAU	1.31354	0.480208	0	16.3875	26.1958						
29.3583	100.357	394.384	88.249	0.921667							
0.450833	0.368438	640.245	1956.47	0	1.66771						
2.17188	54.5302	11.9802	84.5677	8.25938							
0.186146	0.110104	0.0997917	1370.03								
RESEAU	31521.9	0	32532.8	0	5.375	9245.5	0				
8691.96	0	21.1667	6857.42	0	6591.13	0					
17.375	8054.83	0	7680.42	0	21.4583						
9260.96	0	8777.33	0	27.125							

RESEAU	38125.8	0.0416667	38934.1	0	13.5417	
	9912.38	0	9315.54	0	18.6667	7445.25 0
	7190.88	0	17.625	9502.92	0	8966.88 0
	21.25	11534.5	0	10772.9	0	23.3333
RESEAU	41223.5	0	44117.3	0	7.66667	11416.8
	0	10348.1	0	18.2083	7404.46	0 7163.88
	0	21.2083	10571.5	0	9760.63	0 24.0833
	10133.7	0	9617	0	27.375	
RESEAU	24072.1	0	23811.3	0	2.08333	6134.75
	625.221	5822.24	0.0875	14.4833	6279.6	
	0.0208333	6072.88	0.333333	23.2083	6018.93	
	0.00833333	5863	0.308333	27.5708	6236.18	0
	6064.67	0.3	26.6125			
RESEAU	23574.3	0	23412	0	2.08333	6241.08
	0	6073.63	0	16.6667	6340.79	0 6168.42
	0	23.7083	6273.04	0	6108.17	0 25.2083
	6500.63	0	6316.88	0	28.4167	
RESEAU	29324.5	0	31767.7	0	0	1470.38 0
	1253.21	0	0.375	7438.67	0	7355.92 0 1.125
	6652.88	0	6137.96	0	1.04167	7054.46 0
	6653.13	0	1.16667			
RESEAU	71226.3	0	77206	0	14.5833	9857.63
	0	9035.46	0	16.1667	11060.2	0 9442.08
	0	17.4583	12401	0	11294.5	0 17.875
	40514.5	0	39343.7	0	17.125	
RESEAU	88349.4	0	91576.3	0.1	5.17083	9303.49
	0.0375	8418.92	1.575	14.425	11258.1	0 10364
	1.05833	12.2917	7976.13	0.0458333	7264.29	
	2.27917	16.8333	56923.1	0	56464.5	1.125
	13.9583					
RESEAU	85417.3	0	88410.3	0	10.875	11996
	0	11036.4	0	17.6667	8321.5	0 7875.29
	0	13.0833	10598.6	0	9266.96	0 17.7083
	169284	0	166813	1.125	18.2083	
RESEAU	137027	0	138418	0.0833333	7.875	6294.71
	0	5983.04	0.208333	14.1667	5917.21	0
	7448.46	0	12.3333	6408.54	0	5974.79

0.166667	8.25	119189	0.0416667	118392	0.291667
9.54167					
RESEAU	135407	0	135847	0	4.70833
	6440.38	0	13.375	6890.08	0
	8.58333	5913.92	0	5706.67	0
	119780	0	119440	0	6.75
RESEAU	129415	0	129684	0	3.33333
	6003.96	0	13.9167	5568.13	0
	9.08333	5573.5	0	5398.46	0
	114779	0	114464	0	6.875
RESEAU	135304	0	136191	0	0.125 7458 0
	6452.17	0	0.25 4281.33	0	4017.17 0
	0.291667	2078.63	0	1914.54	0 0.416667
	123975	0	123373	0	0

H.9. FICHIER AUXB.DAT

CPU	2.325	0.375	0	32.525	49.325	18.35	112.2	597.7		
	114.725	0.6825	0.3025	0.245	0	0	0	0		
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
CPU	2.8	0	0	34.075	50.625	15.4	108.95			
	639.975	116	0.6775	0.3125	0.2575	0	0			
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0								
CPU	3.225	0.275	0	31.3	51.7	17.025	110.775	617.6		
	115.625	0.685	0.2725	0.235	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	
CPU	2.25	0.3	0	37.3	43.45	19.275	100.675	566.825		
	104.6	0.6875	0.3	0.22	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0		
CPU	2.675	0.375	0	31.6	49.225	19.175	107.775			
	612.775	115.225	0.6725	0.3125	0.26	0	0			
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0								
CPU	2.175	0	0	32	49.75	18.3	108.55	623.25	115.7	
	0.515	0.3125	0.25	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0

	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0			
CPU	3.025	0.4	0	31.175	48.075	20.9	103.8	639.45			
	111.875		0.665	0.3075	0.2775	0	0	0	0		
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
CPU	2.95	0.225	0	32.025	49.65	18.45	105.2	634.125			
	114.55		0.725	0.325	0.25	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0		
CPU	3.15	0.625	0	28.7	51.2	20.225	127.15	538.275			
	123.125		1.43	0.625	0.435	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0		
CPU	1.825	0.75	0	30.125	48.75	21.3	124.25	550.125			
	121.925		0.97	0.6175	0.5375	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	

.
.
.

DISQUE	1.775	6.325	1.05	3.3	6.925	2.05	94	91	35	1
72	25									
DISQUE	0.8	6.475	1.025	2.8	7.175	2.15	94	91	35	1
72	25									
DISQUE	2	5.95	1	4.425	3.65	1.05	94	91	35	1
72	25									

DISQUE	2.7	10.5	1.75	2.5	4.05	1.175	94	91	33.5	1
72	25									
DISQUE	1.775	5.3	0.875	1.45	2.775	0.675	94	91	32	1
72	25									
DISQUE	1.325	8.35	1.375	3.05	5.3	1.475	94	91	32	1
72	25									
DISQUE	0.875	6.8	1.05	1.875	8.95	2.675	94	91	32	1
72	25									
DISQUE	0.85	6.65	1.125	0.875	6.25	1.85	94	91	32	1
72	25									
DISQUE	1.925	23.225		3.5	10.475		22.325		5.95	94
91	32.25	1	72	25						
DISQUE	2.5	12.4	1.875	7.975	20.7	5.9	94	91	31.5	1
72.5	25									

.
.
.

MEMOIRE	2877.35	19650	0	0	0	0	0	0
0	0							
MEMOIRE	3152.5	19600	0	0	0	0	0	0
0	0							
MEMOIRE	3068	19700	0	0	0	0	0	0
0								
MEMOIRE	3163.9	19500	0	0	0	0	0	0
0	0							
MEMOIRE	3173.4	19675	0	0	0	0	0	0
0	0							
MEMOIRE	2978.65	19675	0	0	0	0	0	0
0	0							
MEMOIRE	2825.15	19650	0	0	0	0	0	0
0	0							
MEMOIRE	2844.9	19625	0	0	0	0	0	0
0	0							
MEMOIRE	3121.3	19450	0	0	0	0	0	0
0	0							

MEMOIRE	3141.7	19500	0	0	0	0	0	0
0	0							

.

.

.

PAGE	0	0	0	0	7086	8062.1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	
PAGE	0	0	0	0	7086	8060.9	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	
PAGE	0	0	0	0	7086	8059.6	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	
PAGE	0	0	0	0	7085	8058.4	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	
PAGE	0	0	0	0	7085	8057.1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	
PAGE	0	0	0	0	7085	8055.9	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	
PAGE	0	0	0	0	7084	8054.7	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	
PAGE	0	0	0	0	7084	8053.4	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	
PAGE	0	0	0	0	7080.5	8044.1	0	0	
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
PAGE	0	0	0	0	7058	8016.38	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	

RESEAU	20098	0	20200	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0			
RESEAU	20202	0	20311	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0			
RESEAU	20928	0	21295	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0			
RESEAU	20168	0	20579	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0			
RESEAU	20263	0	20367	0	2	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0			
RESEAU	21047	0	21152	0	3	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0			
RESEAU	20564	0	20671	0	3	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0			
RESEAU	19983	0	20079	0	7	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0			
RESEAU	29288	0	32009	0	42	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0			
RESEAU	39321	0	42372	0	8	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0			

ANNEXE I LES FICHIERS INIT*.DAT
--

I.1. FICHER INIT1.DAT

350804	0	335353	0	5216	59744	20212	1902
4815	1299279	21361068	19053125				

I.2. FICHER INIT2.DAT

667480	0	645969	14	5125	62764	18774	314
3824	1120933	40434661	4150465				

I.3. FICHER INIT3.DAT

386524	0	366123	1	4086	60716	19510	599
3801	886499	22720625	3639309				

I.4. FICHER INIT4.DAT

819647	0	758497	1	8570	289900	137572	15630
39817	3986848	39283293	39705511				

I.5. FICHER INITE.DAT

1284162	0	1305437	0	451	32538	0	0	0
2528448	24001433	5223956						

ANNEXE J LES FICHIERS COMP*.DAT

J.1. FICHER COMPWS1.DAT

0.2	0	0	1.1	2.1	96.8	101132	89460	6284	0.16	0.10	0.12
5596	4619.1										
0	70	0	0	4556	7259.5						
1708	0	1643	0	46							
0.5	0.5	0	1.4	3.2	95.4	88312	78604	5485	0.28	0.15	0.13
5456.8	4675.6										
0	70	0	0	4550	7257.0						
1511	0	1463	0	41							
0	0.8	0	1.1	3.1	95.8	95956	84996	5946	0.33	0.16	0.13
5737	4597.6										
0	70	0	0	4542	7254.0						
1618	0	1554	0	39							
0	0.6	0	1.1	2.9	96	94384	83628	5844	0.17	0.09	0.09
5605.6	4614.5										
0	70	0	0	4535	7251.4						
1606	0	1550	0	49							

J.2. FICHER COMPWS2.DAT

0	0	0	1	0.5	98.5	99800	7806	2700	0.32	0.17	0.13
1513.8	3631.4										
0	56	0	0	4528	8019.3						
1742	0	1644	0	13							
0	0	0	1.1	1.4	97.5	86952	7890	2250	0.03	0.05	0.08
1448.6	3646.4										
0	84	0	0	4527	8016.2						
1537	0	1461	0	15							
0	0.7	0	1	1.1	97.9	93992	7592	2340	0.37	0.13	0.08
1721	3595.7										
0	70	0	0	4526	8013.1						
1631	0	1570	0	13							
0	0.4	0	1	1	98	92208	7614	2274	0.12	0.06	0.07
1715.2	3603.8										

0	70	0	0	4526	8010.0
1592	0	1530	0	28	

J.3. FICHER COMPWS3.DAT

0	0	0	0.9	0.9	98.2	99144	7441	2423	0.16	0.10	0.11
1552.8		5071.6									
0	70	0	0	7785	44310.9						
1651	0	1583	0	14							
0	0	0	1.1	3.1	95.8	86800	6952	2144	0.02	0.07	0.09
1514.6		5050.5									
0	70	0	0	7768	44216.8						
1479	0	1430	0	21							
0	0.4	0	0.9	2.9	96.2	93716	7227	2296	0.22	0.12	0.10
2030.8		5036.6									
0	70	0	0	7747	44123.1						
1600	0	1537	0	26							
0.5	0.2	0	0.9	3	96.1	92552	7315	2258	0.19	0.11	0.09
1773	5049.4										
0	70	0	0	7732	44048.4						
1566	0	1506	0	18							

J.4. FICHER COMPWS4.DAT

0	0	0	1.9	1.7	96.4	100864	88968	6239	0.18	0.11	0.12
5906.2		4530.2									
0	62	0	0	1883	2475.2						
1679	0	1614	0	52							
0	0.5	0	1.8	2.1	96.1	88872	79320	5532	0.44	0.19	0.15
6005	4551.5										
0	90	0	0	1882	2474.8						
1512	0	1462	0	36							
0	0.3	0	2	2	96	95720	84776	5902	0.21	0.11	0.12
6195	4484.6										
0	76	0	0	1881	2474.5						
1621	0	1560	0	25							
0	0.4	0	2	2.1	95.9	94280	83536	5848	0.11	0.13	0.12
6198.2		4510.2									

0	76	0	0	1880	2474.2
1602	0	1541	0	21	

J.5. FICHER COMPELECTRE.DAT

0.4	0	0	4.1	45.3	49.1	112240	18560	11012	0.87	0.34	0.31
0.1	14.1	2.1	0.5	19.3	6	95	94	43	0	95	45
2431.8		15400									
0	98	0	0	2421	3722.1						
5952	0	5928	0	5							
0.4	0	0	8.8	19.2	70.4	98880	17664	9568	0.53	0.31	0.26
0.8	2.2	0.3	4.3	3.9	1	95	94	43	0	95	45
2240.8		15800									
0	98	0	0	2418	3720.3						
5477	0	5481	0	5							
0.7	1.2	0	4.8	32.7	60.9	119600	35328	14160	1.13	0.45	0.29
2.9	6.9	1.6	19.6	97.6	14.8	95	94	43	0	95	45
3082.6		15000									
0	500	0	0	2415	3714.3						
8387	0	12818		0	7						
0	0.8	0	4.2	7.8	86.4	107392	29056	11884	0.77	0.31	0.26
0.3	17.2	2.9	4.8	4.8	1.4	95	94	43	0	95	45
2712.6		15000									
0	152	0	0	2412	3711.4						
6243	0	6299	0	8							

ANNEXE K

LES FICHIERS HEURE.DAT ET DATE.DAT

K.1. FICHER HEURE.DAT

CPU	1.15	0.125	0	0.6	22.225	77.125	414792	75857
	37493		0.9375		0.3525	0.245	0.35	0
	2.175	96.8	378124		336546	22269	0.34	0.145
	0.25	0	0	0.9	0.5	98.6	371156	29443
	0.3425		0.1325		0.0825	0.25	0	0
	98.675		370694		28272	7899	0.1225	0.055
	0.15	0	0	1.225	2.35	96.425	378296	336464
	22236		0.2525		0.105	0.085		
DISQUE	0.675	3.325	0.6		1.275	5.75	1.75	97
	92	93						94
								19
								0
MEMOIRE	2612.55		16000		5775.95	4667.9	2189.95	
	1344.55		2181.95		5331.5	6416.65	4518.25	
PAGE	0	392	0	0	59956.3	25784.8	0	280
	0	0	10670.3		21034.4	0	266	0
	23199.4		0	280	0	0	10447.5	22041.8
	294	0	0	3616	3525.8			0
RESEAU	20175		0	20165	0	1	5612	0
	0	31	5668	0	5533	0	13	5537
	24	5596	0	5457	0	35	0	5396
								0

K.2. FICHER DATE.DAT

Tue May 1 00:59:03 WET DST 1990

ANNEXE L LA CRONTAB

1,15,30,45 * * * 2-7	dedo/bin/sh/usr/users/inf/dedo/memoire/wsun
1,15,30,45 * * * 2-7	dedo/bin/sh/usr/users/inf/dedo/memoire/wsdeux
1,15,30,45 * * * 2-7	dedo/bin/sh/usr/users/inf/dedo/memoire/wstois
1,15,30,45 * * * 2-7	dedo/bin/sh/usr/users/inf/dedo/memoire/wsquatre
1,15,30,45 * * * 2-7	dedo/bin/sh /usr/users/inf/dedo/memoire/electre
8,23,38,53 * * * 2-7	dedo/bin/sh/usr/users/inf/dedo/memoire/compws1
8,23,38,53 * * * 2-7	dedo/bin/sh/usr/users/inf/dedo/memoire/compws2
8,23,38,53 * * * 2-7	dedo/bin/sh/usr/users/inf/dedo/memoire/compws3
8,23,38,53 * * * 2-7	dedo/bin/sh/usr/users/inf/dedo/memoire/compws4
8,23,38,53 * * * 2-7	dedo/bin/sh/usr/users/inf/dedo/memoire/compelectre
55 * * * 2-7	dedo /bin/sh /usr/users/inf/dedo/memoire/compheure
57 0 * * *	dedo /bin/sh /usr/users/inf/dedo/memoire/compjour
59 0 1 * *	dedo /bin/sh/usr/users/inf/dedo/memoire/compmois

1,15,30,45 0-7 * * 1	dedo /bin/sh/usr/users/inf/dedo/memoire/wsun
1,15,30,45 12-23 * * 1	dedo /bin/sh/usr/users/inf/dedo/memoire/wsun
1,15,30,45 0-7 * * 1	dedo / ... /wsdeux
1,15,30,45 12-23 * * 1	dedo / ... /wsdeux
1,15,30,45 0-7 * * 1	dedo / ... /wstois
1,15,30,45 12-23 * * 1	dedo / ... /wstois
1,15,30,45 0-7 * * 1	dedo / ... /wsquatre
1,15,30,45 12-23 * * 1	dedo / ... /wsquatre
1,15,30,45 0-7 * * 1	dedo / ... /electre
1,15,30,45 12-23 * * 1	dedo / ... /electre

8,23,38,53 0-7 * * 1	dedo / ... /compws1
8,23,38,53 12-23 * * 1	dedo / ... /compws1
8,23,38,53 0-7 * * 1	dedo / ... /compws2
8,23,38,53 12-23 * * 1	dedo / ... /compws2
8,23,38,53 0-7 * * 1	dedo / ... /compws3
8,23,38,53 12-23 * * 1	dedo / ... /compws3
8,23,38,53 0-7 * * 1	dedo / ... /compws4
8,23,38,53 12-23 * * 1	dedo / ... /compws4
8,23,38,53 0-7 * * 1	dedo / ... /compelectre
8,23,38,53 12-23 * * 1	dedo / ... /compelectre
55 0-7 * * 1	dedo / ... /compheure
55 12-23 * * 1	dedo / ... /compheure
58 7 * * 1	dedo / ... /compbackup

ANNEXE M

LA MACRO-COMMANDE EXCEL

=AJOUTER.BARRE()
=AJOUTER.MENU(A1;A25:D30)
=AJOUTER.MENU(A1;A32:D37)
=AJOUTER.MENU(A1;A39:D44)
=AJOUTER.MENU(A1;A46:D50)
=AJOUTER.MENU(A1;A52:D56)
=AFFICHER.BARRE(A1)
=FORMULE(0;A23)
=FORMULE(0;B23)
=FORMULE(0;C23)
=FORMULE(4;D23)
=FORMULE(1;E23)
=FORMULE(1;F23)
=OUVRIR("performance.mc";FAUX;FAUX;1;"MP")
=SI(CELLULE.ACTIVE()="CPUX";ATTEINDRE(A20))
=REEMPLACER.CELLULE(":";";")
=SELECTIONNER("L1C1")
=FORMULE("CPUX")
=ENREGISTRER.SOUS(INDEX(FENETRES();1);1;"MP");)
=FORMULE("performance.mc";A22)
=ARRETER()
performance.mc
0
Fichier
Enregistrer ...
Imprimer ...
-
Quitter Performance
Quitter EXCEL
Période
Jour courant
Jour précédent
-
Mois courant
Mois précédent
Données
CPU ...
Disque ...
Mémoire
Pagination ...
Réseau ...
Hôtes
Stations VAX ...
Stations SUN ...
Station HP ...
Serveur JUNON ...
AFFICHAGE
Tableau des résultats
-
Graphique des résultats
Graphique des moyennes
Impr
=ZONE.DE.DIALOGUE(B197:H207)
=SI(A59=FAUX;RETOUR())
=IMPRIMER(1;1;1;H200;;H205;;H204;H201)
=RETOUR()
Quitter
=QUITTER()
=RETOUR()
Quitterp
=AFFICHER.BARRE(1)
=RETOUR()

Enreg
=SI(INDEX(FENETRES();1)="performance.mc";ATTEINDRE(A265))
=SI(INDEX(FENETRES();1)="performance.mp";ATTEINDRE(A265))
=SI(INDEX(FENETRES();1)="performance.jc";ATTEINDRE(A265))
=SI(INDEX(FENETRES();1)="performance.jp";ATTEINDRE(A265))
=SELECTION.ATTEINDRE(H191)
=EFFACER(1)
=SELECTION.ATTEINDRE(H193)
=EFFACER(1)
=ZONE.DE.DIALOGUE(B188:H195)
=SI(A81=FAUX;RETOUR())
=SI(H191="";ATTEINDRE(A208))
=SI(H193="";ATTEINDRE(A270))
=ACTIVER.SUIV()
=ENREGISTRER.SOUS(H191;;H193;FAUX)
=RETOUR()
jourc
=FORMULE(1;F23)
=SI(NON(A23=0);COMMANDE.COCHER(A1;2;A23;FAUX))
=FORMULE(1;A23)
=COMMANDE.COCHER(A1;2;1;VRAI)
=RETOUR()
jourp
=FORMULE(1;F23)
=SI(NON(A23=0);COMMANDE.COCHER(A1;2;A23;FAUX))
=FORMULE(2;A23)
=COMMANDE.COCHER(A1;2;2;VRAI)
=RETOUR()
moisc
=FORMULE(1;F23)
=SI(NON(A23=0);COMMANDE.COCHER(A1;2;A23;FAUX))
=FORMULE(4;A23)
=COMMANDE.COCHER(A1;2;4;VRAI)
=RETOUR()
moisp
=FORMULE(1;F23)
=SI(NON(A23=0);COMMANDE.COCHER(A1;2;A23;FAUX))
=FORMULE(5;A23)
=COMMANDE.COCHER(A1;2;5;VRAI)
=RETOUR()
cpu
=ZONE.DE.DIALOGUE(B220:H230)
=SI(A118=FAUX;RETOUR())
=FORMULE(1;F23)
=SI(NON(B23=0);COMMANDE.COCHER(A1;3;B23;FAUX))
=FORMULE(1;B23)
=COMMANDE.COCHER(A1;3;1;VRAI)
=RETOUR()
disque
=ZONE.DE.DIALOGUE(B232:H242)
=SI(A127=FAUX;RETOUR())
=FORMULE(1;F23)
=SI(NON(B23=0);COMMANDE.COCHER(A1;3;B23;FAUX))
=FORMULE(2;B23)
=COMMANDE.COCHER(A1;3;2;VRAI)
=SI(ET(H238=FAUX;H239=FAUX;H240=FAUX);ATTEINDRE(A182))
=RETOUR()
mémoire
=FORMULE(1;F23)
=SI(NON(B23=0);COMMANDE.COCHER(A1;3;B23;FAUX))
=FORMULE(3;B23)
=COMMANDE.COCHER(A1;3;3;VRAI)
=RETOUR()

page
=ZONE.DE.DIALOGUE(B253:H262)
=SI(A144=FAUX;RETOUR())
=FORMULE(1;F23)
=SI(NON(B23=0);COMMANDE.COCHER(A1;3;B23;FAUX))
=FORMULE(4;B23)
=COMMANDE.COCHER(A1;3;4;VRAI)
=RETOUR()
réseau
=ZONE.DE.DIALOGUE(B244:H251)
=SI(A153=FAUX;RETOUR())
=FORMULE(1;F23)
=SI(NON(B23=0);COMMANDE.COCHER(A1;3;B23;FAUX))
=FORMULE(5;B23)
=COMMANDE.COCHER(A1;3;5;VRAI)
=RETOUR()
vax
=ZONE.DE.DIALOGUE(B209:H218)
=SI(A162=FAUX;RETOUR())
=FORMULE(1;F23)
=SI(NON(C23=0);COMMANDE.COCHER(A1;4;C23;FAUX))
=FORMULE(1;C23)
=COMMANDE.COCHER(A1;4;1;VRAI)
=RETOUR()
sun
=ALERTE("Les mesures pour les stations Sun ne sont pas encore disponibles.";2)
=RETOUR()
hp
=ALERTE("Les mesures pour la station HP ne sont pas encore disponibles.";2)
=RETOUR()
junon
=ALERTE("Les mesures pour le serveur Junon ne sont pas encore disponibles.";2)
=RETOUR()
=ALERTE("Il faut cocher au moins une des cases !";2)
=ATTEINDRE(A126)
=ALERTE("Seul ELECTRE possède des disques ";2)
=ARRETER()
ZONE DE DIALOGUE
Texte fixe
Texte fixe
Zone d'entree de texte
Texte fixe
Zone d'entree de texte
Bouton OK
Bouton Annuler
ZONE DE DIALOGUE
Texte fixe
Texte fixe
Modifier texte
Zone de groupe
Bouton option
Bouton option
Case à cocher
Case à cocher
Bouton OK
Bouton Annuler
ZONE DE DIALOGUE
texte fixe
Zone de groupe
Bouton option
Bouton option

Bouton option
Bouton option
Bouton option
Bouton Annuler
Bouton OK
ZONE DE DIALOGUE
texte fixe
Zone de groupe
Bouton option
Bouton option
Bouton option
Bouton option
Bouton option
Bouton option
Bouton Annuler
Bouton OK
ZONE DE DIALOGUE
texte fixe
Zone de groupe
Bouton option
Bouton option
Bouton option
Case à cocher
Case à cocher
Case à cocher
Bouton Annuler
Bouton OK
ZONE DE DIALOGUE
texte fixe
Zone de groupe
Bouton option
Bouton option
Bouton option
Bouton Annuler
Bouton OK
ZONE DE DIALOGUE
texte fixe
Zone de groupe
Bouton option
Bouton option
Bouton option
Bouton option
Bouton Annuler
Bouton OK
=ALERTE("Vous ne pouvez pas enregistrer les données brutes";3)
=ARRETER()
=ALERTE("Il faut donner un nom de fichier !";2)
=ARRETER()
=ALERTE("Il faut donner un mot de passe non vide !";2)
=ARRETER()
=ALERTE("Il faut sélectionner une période";2)
=ARRETER()
=ALERTE("Il faut sélectionner un type de données !";2)
=ARRETER()
=ALERTE("Il faut sélectionner un hôte !";2)
=ARRETER()
table
=SI(A23=0;ATTEINDRE(A273))
=SI(B23=0;ATTEINDRE(A276))

=SI(C23=0;ATTEINDRE(A279))
=SI(A23=D23;ATTEINDRE(A299))
=FORMULE(A23;D23)
=FERMER(FAUX)
=SI(A23=1;FORMULE("performance.jc";A22))
=SI(A23=2;FORMULE("performance.jp";A22))
=SI(A23=4;FORMULE("performance.mc";A22))
=SI(A23=5;FORMULE("performance.mp";A22))
=OUVRIR(A22;FAUX;FAUX;"MP")
=SI(CELLULE.ACTIVE()="CPUX";ATTEINDRE(A299))
=REPLACER.CELLULE(";",",")
=SELECTIONNER("L1C1")
=FORMULE("CPUX")
=ENREGISTRER.SOUS(INDEX(FENETRES();1);1;"MP");)
=SI(H211=1;FORMULE("ELECTRE";C22))
=SI(H211=2;FORMULE("WS1";C22))
=SI(H211=3;FORMULE("WS2";C22))
=SI(H211=4;FORMULE("WS3";C22))
=SI(H211=5;FORMULE("WS4";C22))
=SI(B23<>1;ATTEINDRE(A365))
=SI(H222<>1;ATTEINDRE(A316))
=SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L1C2:L31C4";B22);FORMULE("L1C2:L24C4";B22)))
=SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L1C14:L31C16";B22);FORMULE("L1C14:L24C16";B22)))
=SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L1C26:L31C28";B22);FORMULE("L1C26:L24C28";B22)))
=SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L1C38:L31C40";B22);FORMULE("L1C38:L24C40";B22)))
=SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L1C50:L31C52";B22);FORMULE("L1C50:L24C52";B22)))
=FORMULE("Etat des processus";D22)
=FORMULE("3";E22)
=FORMULE("Exécuté";F22)
=FORMULE("Bloqué";G22)
=FORMULE("En attente";H22)
=SI(H222<>2;ATTEINDRE(A327))
=SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L1C5:L31C7";B22);FORMULE("L1C5:L24C7";B22)))
=SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L1C17:L31C19";B22);FORMULE("L1C17:L24C19";B22)))
=SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L1C29:L31C31";B22);FORMULE("L1C29:L24C31";B22)))
=SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L1C41:L31C43";B22);FORMULE("L1C41:L24C43";B22)))
=SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L1C53:L31C55";B22);FORMULE("L1C53:L24C55";B22)))
=FORMULE("Occupation du CPU";D22)
=FORMULE("3";E22)
=FORMULE("Utilisateur";F22)
=FORMULE("Système";G22)
=FORMULE("Endormi";H22)
=SI(H222<>3;ATTEINDRE(A336))
=SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L1C8:L31C8";B22);FORMULE("L1C8:L24C8";B22)))
=SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L1C20:L31C20";B22);FORMULE("L1C20:L24C20";B22)))
=SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L1C32:L31C32";B22);FORMULE("L1C32:L24C32";B22)))
=SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L1C44:L31C44";B22);FORMULE("L1C44:L24C44";B22)))
=SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L1C56:L31C56";B22);FORMULE("L1C56:L24C56";B22)))
=FORMULE("Nombre d'interruptions";D22)
=FORMULE("1";E22)
=FORMULE("Interruptions";F22)
=SI(H222<>4;ATTEINDRE(A345))
=SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L1C9:L31C9";B22);FORMULE("L1C9:L24C9";B22)))
=SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L1C21:L31C21";B22);FORMULE("L1C21:L24C21";B22)))
=SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L1C33:L31C33";B22);FORMULE("L1C33:L24C33";B22)))
=SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L1C45:L31C45";B22);FORMULE("L1C45:L24C45";B22)))
=SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L1C57:L31C57";B22);FORMULE("L1C57:L24C57";B22)))
=FORMULE("Appels système";D22)
=FORMULE("1";E22)
=FORMULE("nombre";F22)
=SI(H222<>5;ATTEINDRE(A354))
=SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L1C10:L31C10";B22);FORMULE("L1C10:L24C10";B22)))
=SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L1C22:L31C22";B22);FORMULE("L1C22:L24C22";B22)))
=SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L1C34:L31C34";B22);FORMULE("L1C34:L24C34";B22)))
=SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L1C46:L31C46";B22);FORMULE("L1C46:L24C46";B22)))
=SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L1C58:L31C58";B22);FORMULE("L1C58:L24C58";B22)))
=FORMULE("Changements de contexte";D22)
=FORMULE("1";E22)
=FORMULE("nombre";F22)
=SI(H222<>6;ATTEINDRE(A365))
=SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L1C11:L31C13";B22);FORMULE("L1C11:L24C13";B22)))

=SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L1C23:L31C25";B22);FORMULE("L1C23:L24C25";B22)))
 =SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L1C35:L31C37";B22);FORMULE("L1C35:L24C37";B22)))
 =SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L1C47:L31C49";B22);FORMULE("L1C47:L24C49";B22)))
 =SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L1C59:L31C61";B22);FORMULE("L1C59:L24C61";B22)))
 =FORMULE("Charge moyenne";D22)
 =FORMULE("3";E22)
 =FORMULE("1 minute";F22)
 =FORMULE("5 minutes";G22)
 =FORMULE("15 minutes";H22)
 =SI(B23<>2;ATTEINDRE(A437))
 =SI(H211<>1;ATTEINDRE(A185))
 =SI(H234=3;ATTEINDRE(A431))
 =SI(NON(ET(H238=VRAI;H239=VRAI;H240=VRAI));ATTEINDRE(A373))
 =FORMULE("Nb Ops/Sec";F22)
 =FORMULE("Nb Bytes/Sec";G22)
 =FORMULE("Temps d'accès";H22)
 =FORMULE(3;E22)
 =SI(NON(ET(H238=VRAI;H239=VRAI;H240=FAUX));ATTEINDRE(A377))
 =FORMULE("Nb Ops/Sec";F22)
 =FORMULE("Nb Bytes/Sec";G22)
 =FORMULE(2;E22)
 =SI(NON(ET(H238=VRAI;H239=FAUX;H240=VRAI));ATTEINDRE(A381))
 =FORMULE("Nb Ops/Sec";F22)
 =FORMULE("Temps d'accès";G22)
 =FORMULE(2;E22)
 =SI(NON(ET(H238=VRAI;H239=FAUX;H240=FAUX));ATTEINDRE(A384))
 =FORMULE("Nb Ops/Sec";F22)
 =FORMULE(1;E22)
 =SI(NON(ET(H238=FAUX;H239=VRAI;H240=VRAI));ATTEINDRE(A388))
 =FORMULE("Nb Bytes/Sec";F22)
 =FORMULE("Temps d'accès";G22)
 =FORMULE(2;E22)
 =SI(NON(ET(H238=FAUX;H239=VRAI;H240=FAUX));ATTEINDRE(A391))
 =FORMULE("Nb Bytes/Sec";F22)
 =FORMULE(1;E22)
 =SI(NON(ET(H238=FAUX;H239=FAUX;H240=VRAI));ATTEINDRE(A394))
 =FORMULE("Temps d'accès";F22)
 =FORMULE(1;E22)
 =SI(H234<>1;ATTEINDRE(A412))
 =SI(ET(H238=VRAI;H239=VRAI;H240=VRAI);SI(OU(D23=4;D23=5);FORMULE("L32C2:L62C4";B22);FO
 =SI(ET(H238=VRAI;H239=VRAI;H240=FAUX);SI(OU(D23=4;D23=5);FORMULE("L32C2:L62C3";B22);FO
 =SI(ET(H238=VRAI;H239=FAUX;H240=VRAI);SI(OU(D23=4;D23=5);ATTEINDRE(A404);ATTEINDRE(A-
 =SI(ET(H238=VRAI;H239=FAUX;H240=FAUX);SI(OU(D23=4;D23=5);FORMULE("L32C2:L62C2";B22);FO
 =SI(ET(H238=FAUX;H239=VRAI;H240=VRAI);SI(OU(D23=4;D23=5);FORMULE("L32C3:L62C4";B22);FO
 =SI(ET(H238=FAUX;H239=VRAI;H240=FAUX);SI(OU(D23=4;D23=5);FORMULE("L32C3:L62C3";B22);FO
 =SI(ET(H238=FAUX;H239=FAUX;H240=VRAI);SI(OU(D23=4;D23=5);FORMULE("L32C4:L62C4";B22);FO
 =FORMULE("Accès au disque 1";D22)
 =ATTEINDRE(A437)
 =FORMULE(0;B22)
 =FORMULE("L32C2:L62C2";I22)
 =FORMULE("L32C4:L62C4";J22)
 =ATTEINDRE(A402)
 =FORMULE(0;B22)
 =FORMULE("L25C2:L48C2";I22)
 =FORMULE("L25C4:L48C4";J22)
 =ATTEINDRE(A402)
 =SI(H234<>2;ATTEINDRE(A430))
 =SI(ET(H238=VRAI;H239=VRAI;H240=VRAI);SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L32C5:L62C7
 =SI(ET(H238=VRAI;H239=VRAI;H240=FAUX);SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L32C5:L62C
 =SI(ET(H238=VRAI;H239=FAUX;H240=VRAI);SI(H211=1;SI(OU(D23=4;D23=5);ATTEINDRE(A422);ATT
 =SI(ET(H238=VRAI;H239=FAUX;H240=FAUX);SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L32C5:L62C
 =SI(ET(H238=FAUX;H239=VRAI;H240=VRAI);SI(OU(D23=4;D23=5);FORMULE("L32C6:L62C
 =SI(ET(H238=FAUX;H239=VRAI;H240=FAUX);SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L32C6:L62C
 =SI(ET(H238=FAUX;H239=FAUX;H240=VRAI);SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L32C7:L62C
 =FORMULE("Accès au disque 2";D22)
 =ATTEINDRE(A437)
 =FORMULE(0;B22)
 =FORMULE("L32C5:L62C5";I22)
 =FORMULE("L32C7:L62C7";J22)
 =ATTEINDRE(A420)
 =FORMULE(0;B22)

=FORMULE("L25C5:L48C5";I22)
 =FORMULE("L25C7:L48C7";J22)
 =ATTEINDRE(A420)
 =SI(H234<>3;ATTEINDRE(A437))
 =FORMULE("Partitions disque";D22)
 =FORMULE(6;E22)
 =FORMULE("Racine";F22)
 =FORMULE("Usr";G22)
 =FORMULE("Var";H22)
 =SI(OU(D23=4;D23=5);FORMULE("L32C8:L62C13";B22);FORMULE("L25C8:L48C13";B22))
 =SI(B23<>3;ATTEINDRE(A447))
 =SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L63C2:L93C3";B22);FORMULE("L49C2:L72C3";B22)))
 =SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L63C4:L93C5";B22);FORMULE("L49C4:L72C5";B22)))
 =SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L63C6:L93C7";B22);FORMULE("L49C6:L72C7";B22)))
 =SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L63C8:L93C9";B22);FORMULE("L49C8:L72C9";B22)))
 =SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L63C10:L93C11";B22);FORMULE("L49C10:L72C11";B22)))
 =FORMULE("Utilisation de la mémoire";D22)
 =FORMULE(2;E22)
 =FORMULE("Virtuelle util.";F22)
 =FORMULE("Libre";G22)
 =SI(B23<>4;ATTEINDRE(A494))
 =SI(H255<>1;ATTEINDRE(A457))
 =SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L94C2:L124C2";B22);FORMULE("L73C2:L96C2";B22)))
 =SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L94C8:L124C8";B22);FORMULE("L73C8:L96C8";B22)))
 =SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L94C14:L124C14";B22);FORMULE("L73C14:L96C14";B22)))
 =SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L94C20:L124C20";B22);FORMULE("L73C20:L96C20";B22)))
 =SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L94C26:L124C26";B22);FORMULE("L73C26:L96C26";B22)))
 =FORMULE("Défauts de page";D22)
 =FORMULE("1";E22)
 =FORMULE("nombre";F22)
 =SI(H255<>2;ATTEINDRE(A466))
 =SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L94C3:L124C3";B22);FORMULE("L73C3:L96C3";B22)))
 =SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L94C9:L124C9";B22);FORMULE("L73C9:L96C9";B22)))
 =SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L94C15:L124C15";B22);FORMULE("L73C15:L96C15";B22)))
 =SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L94C21:L124C21";B22);FORMULE("L73C21:L96C21";B22)))
 =SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L94C27:L124C27";B22);FORMULE("L73C27:L96C27";B22)))
 =FORMULE("Pages en entrée";D22)
 =FORMULE("1";E22)
 =FORMULE("nombre";F22)
 =SI(H255<>3;ATTEINDRE(A475))
 =SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L94C4:L124C4";B22);FORMULE("L73C4:L96C4";B22)))
 =SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L94C10:L124C10";B22);FORMULE("L73C10:L96C10";B22)))
 =SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L94C16:L124C16";B22);FORMULE("L73C16:L96C16";B22)))
 =SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L94C22:L124C22";B22);FORMULE("L73C22:L96C22";B22)))
 =SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L94C28:L124C28";B22);FORMULE("L73C28:L96C28";B22)))
 =FORMULE("Pages en sortie";D22)
 =FORMULE("1";E22)
 =FORMULE("nombre";F22)
 =SI(H255<>4;ATTEINDRE(A484))
 =SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L94C5:L124C5";B22);FORMULE("L73C5:L96C5";B22)))
 =SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L94C11:L124C11";B22);FORMULE("L73C11:L96C11";B22)))
 =SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L94C17:L124C17";B22);FORMULE("L73C17:L96C17";B22)))
 =SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L94C23:L124C23";B22);FORMULE("L73C23:L96C23";B22)))
 =SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L94C29:L124C29";B22);FORMULE("L73C29:L96C29";B22)))
 =FORMULE("Pages libérées";D22)
 =FORMULE("1";E22)
 =FORMULE("nombre";F22)
 =SI(H255<>5;ATTEINDRE(A494))
 =SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L94C6:L124C7";B22);FORMULE("L73C6:L96C7";B22)))
 =SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L94C12:L124C13";B22);FORMULE("L73C12:L96C13";B22)))
 =SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L94C18:L124C19";B22);FORMULE("L73C18:L96C19";B22)))
 =SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L94C24:L124C25";B22);FORMULE("L73C24:L96C25";B22)))
 =SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L94C30:L124C31";B22);FORMULE("L73C30:L96C31";B22)))
 =FORMULE("Temps moyen entre pagination";D22)
 =FORMULE("2";E22)
 =FORMULE("entre défauts";F22)
 =FORMULE("entre pages";G22)
 =SI(B23<>5;ATTEINDRE(A524))
 =SI(H246<>1;ATTEINDRE(A505))
 =SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L125C2:L155C3";B22);FORMULE("L97C2:L120C3";B22)))
 =SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L125C7:L155C8";B22);FORMULE("L97C7:L120C8";B22)))

=SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L125C12:L155C13";B22);FORMULE("L97C12:L120C13";B22)))
 =SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L125C17:L155C18";B22);FORMULE("L97C17:L120C18";B22)))
 =SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L125C22:L155C23";B22);FORMULE("L97C22:L120C23";B22)))
 =FORMULE("Paquets en entrée";D22)
 =FORMULE("2";E22)
 =FORMULE("paquets";F22)
 =FORMULE("erreurs";G22)
 =SI(H246<>2;ATTEINDRE(A515))
 =SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L125C4:L155C5";B22);FORMULE("L97C4:L120C5";B22)))
 =SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L125C9:L155C10";B22);FORMULE("L97C9:L120C10";B22)))
 =SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L125C14:L155C15";B22);FORMULE("L97C14:L120C15";B22)))
 =SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L125C19:L155C20";B22);FORMULE("L97C19:L120C20";B22)))
 =SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L125C24:L155C25";B22);FORMULE("L97C24:L120C25";B22)))
 =FORMULE("Paquets en sortie";D22)
 =FORMULE("2";E22)
 =FORMULE("paquets";F22)
 =FORMULE("erreurs";G22)
 =SI(H246<>3;ATTEINDRE(A524))
 =SI(H211=1;SI(OU(D23=4;D23=5);FORMULE("L125C6:L155C6";B22);FORMULE("L97C6:L120C6";B22)))
 =SI(H211=2;SI(OU(D23=4;D23=5);FORMULE("L125C11:L155C11";B22);FORMULE("L97C11:L120C11";B22)))
 =SI(H211=3;SI(OU(D23=4;D23=5);FORMULE("L125C16:L155C16";B22);FORMULE("L97C16:L120C16";B22)))
 =SI(H211=4;SI(OU(D23=4;D23=5);FORMULE("L125C21:L155C21";B22);FORMULE("L97C21:L120C21";B22)))
 =SI(H211=5;SI(OU(D23=4;D23=5);FORMULE("L125C26:L155C26";B22);FORMULE("L97C26:L120C26";B22)))
 =FORMULE("Collisions";D22)
 =FORMULE("1";E22)
 =FORMULE("nombre";F22)
 =NOUVEAU.DOCUMENT(1)
 =SELECTIONNER("L2C1")
 =FORMULE(C22)
 =SELECTIONNER("L2C1")
 =FORMAT.POLICE("Chicago";12;VRAI;FAUX;VRAI;FAUX;0;FAUX;FAUX)
 =SELECTIONNER("L2C2")
 =FORMULE(D22)
 =SELECTIONNER("L2C2")
 =FORMAT.POLICE("Chicago";12;VRAI;FAUX;VRAI;FAUX;0;FAUX;FAUX)
 =SELECTIONNER("L4C2")
 =FORMULE(F22)
 =SI(E22=1;ATTEINDRE(A548))
 =SELECTIONNER("L4C3")
 =FORMULE(G22)
 =SI(E22=2;ATTEINDRE(A548))
 =SELECTIONNER("L4C4")
 =FORMULE(H22)
 =SI(E22=3;ATTEINDRE(A548))
 =SELECTIONNER("L4C5")
 =FORMULE("Tmp")
 =SELECTIONNER("L4C6")
 =FORMULE("Users")
 =SELECTIONNER("L4C7")
 =FORMULE("Students")
 =SELECTIONNER("L4C1:L4C7")
 =FORMAT.POLICE("Bookman";10;VRAI;FAUX;FAUX;FAUX;0;FAUX;FAUX)
 =POSITION(3)
 =SELECTIONNER("L4C1:L4C2")
 =ENCADREMENT(VRAI;FAUX;FAUX;FAUX;FAUX;FAUX)
 =SI(E22=1;ATTEINDRE(A562))
 =SELECTIONNER("L4C1:L4C3")
 =ENCADREMENT(VRAI;FAUX;FAUX;FAUX;FAUX;FAUX)
 =SI(E22=2;ATTEINDRE(A562))
 =SELECTIONNER("L4C1:L4C4")
 =ENCADREMENT(VRAI;FAUX;FAUX;FAUX;FAUX;FAUX)
 =SI(E22=3;ATTEINDRE(A562))
 =SELECTIONNER("L4C1:L4C7")
 =ENCADREMENT(VRAI;FAUX;FAUX;FAUX;FAUX;FAUX)
 =SI(OU(A22="performance.mc";A22="performance.mp");ATTEINDRE(A618))
 =SELECTIONNER("L5C1")
 =FORMULE("01 Hr")
 =SELECTIONNER("L6C1")
 =FORMULE("02 Hr")
 =SELECTIONNER("L7C1")
 =FORMULE("03 Hr")

=SELECTIONNER("L8C1")
=FORMULE("04 Hr")
=SELECTIONNER("L9C1")
=FORMULE("05 Hr")
=SELECTIONNER("L10C1")
=FORMULE("06 Hr")
=SELECTIONNER("L11C1")
=FORMULE("07 Hr")
=SELECTIONNER("L12C1")
=FORMULE("08 Hr")
=SELECTIONNER("L13C1")
=FORMULE("09 Hr")
=SELECTIONNER("L14C1")
=FORMULE("10 Hr")
=SELECTIONNER("L15C1")
=FORMULE("11 Hr")
=SELECTIONNER("L16C1")
=FORMULE("12 Hr")
=SELECTIONNER("L17C1")
=FORMULE("13 Hr")
=SELECTIONNER("L18C1")
=FORMULE("14 Hr")
=SELECTIONNER("L19C1")
=FORMULE("15 Hr")
=SELECTIONNER("L20C1")
=FORMULE("16 Hr")
=SELECTIONNER("L21C1")
=FORMULE("17 Hr")
=SELECTIONNER("L22C1")
=FORMULE("18 Hr")
=SELECTIONNER("L23C1")
=FORMULE("19 Hr")
=SELECTIONNER("L24C1")
=FORMULE("20 Hr")
=SELECTIONNER("L25C1")
=FORMULE("21 Hr")
=SELECTIONNER("L26C1")
=FORMULE("22 Hr")
=SELECTIONNER("L27C1")
=FORMULE("23 Hr")
=SELECTIONNER("L28C1")
=FORMULE("24 Hr")
=SELECTIONNER("L30C1")
=FORMULE("Moyenne")
=SELECTIONNER("L4C1:L30C1")
=FORMAT.POLICE("Bookman";12;VRAI;FAUX;FAUX;FAUX;0;FAUX;FAUX)
=POSITION(3)
=ENCADREMENT(VRAI;;FAUX;;;FAUX)
=SI(OU(A22="performance.jc";A22="performance.jp");ATTEINDRE(A686))
=SELECTIONNER("L5C1")
=FORMULE("01")
=SELECTIONNER("L6C1")
=FORMULE("02")
=SELECTIONNER("L7C1")
=FORMULE("03")
=SELECTIONNER("L8C1")
=FORMULE("04")
=SELECTIONNER("L9C1")
=FORMULE("05")
=SELECTIONNER("L10C1")
=FORMULE("06")
=SELECTIONNER("L11C1")
=FORMULE("07")
=SELECTIONNER("L12C1")
=FORMULE("08")
=SELECTIONNER("L13C1")
=FORMULE("09")
=SELECTIONNER("L14C1")
=FORMULE("10")
=SELECTIONNER("L15C1")
=FORMULE("11")

=SELECTIONNER("L16C1")
=FORMULE("12")
=SELECTIONNER("L17C1")
=FORMULE("13")
=SELECTIONNER("L18C1")
=FORMULE("14")
=SELECTIONNER("L19C1")
=FORMULE("15")
=SELECTIONNER("L20C1")
=FORMULE("16")
=SELECTIONNER("L21C1")
=FORMULE("17")
=SELECTIONNER("L22C1")
=FORMULE("18")
=SELECTIONNER("L23C1")
=FORMULE("19")
=SELECTIONNER("L24C1")
=FORMULE("20")
=SELECTIONNER("L25C1")
=FORMULE("21")
=SELECTIONNER("L26C1")
=FORMULE("22")
=SELECTIONNER("L27C1")
=FORMULE("23")
=SELECTIONNER("L28C1")
=FORMULE("24")
=SELECTIONNER("L29C1")
=FORMULE("25")
=SELECTIONNER("L30C1")
=FORMULE("26")
=SELECTIONNER("L31C1")
=FORMULE("27")
=SELECTIONNER("L32C1")
=FORMULE("28")
=SELECTIONNER("L33C1")
=FORMULE("29")
=SELECTIONNER("L34C1")
=FORMULE("30")
=SELECTIONNER("L35C1")
=FORMULE("31")
=SELECTIONNER("L37C1")
=FORMULE("Moyenne")
=SELECTIONNER("L4C1:L37C1")
=FORMAT.POLICE("Bookman",12;VRAI;FAUX;FAUX;FAUX;0;FAUX;FAUX)
=POSITION(3)
=ENCADREMENT(VRAI;;FAUX;;;FAUX)
=SI(B22=0;ATTEINDRE(A691))
=ACTIVER(A22)
=SELECTIONNER(LIRE.CELLULE(5;B22))
=COPIER()
=ATTEINDRE(A706)
=ACTIVER(A22)
=SELECTIONNER(LIRE.CELLULE(5;I22))
=COPIER()
=ACTIVER.SUIV()
=SELECTIONNER("L5C2")
=COLLER()
=POSITION(3)
=ACTIVER(A22)
=SELECTIONNER(LIRE.CELLULE(5;J22))
=COPIER()
=ACTIVER.SUIV()
=SELECTIONNER("L5C3")
=COLLER()
=POSITION(3)
=ATTEINDRE(A709)
=ACTIVER.SUIV()
=SELECTIONNER("L5C2")
=COLLER()
=SELECTIONNER("L5C2:L37C7")
=POSITION(3)

```

=SI(OU(A22="performance.mc";A22="performance.mp");ATTEINDRE(A728))
=FORMULE("=MOYENNE(L5C2:L28C2)","L30C2")
=SI(E22=1;ATTEINDRE(A721))
=FORMULE("=MOYENNE(L5C3:L28C3)","L30C3")
=SI(E22=2;ATTEINDRE(A721))
=FORMULE("=MOYENNE(L5C4:L28C4)","L30C4")
=SI(E22=3;ATTEINDRE(A721))
=FORMULE("=MOYENNE(L5C5:L28C5)","L30C5")
=FORMULE("=MOYENNE(L5C6:L28C6)","L30C6")
=FORMULE("=MOYENNE(L5C7:L28C7)","L30C7")
=SELECTIONNER("L4")
=COPIER()
=SELECTIONNER("L29")
=COLLER()
=POSITION(3)
=FORMAT.POLICE("Bookman";12;VRAI;FAUX;FAUX;FAUX;0;FAUX;FAUX)
=ATTEINDRE(A743)
=FORMULE("=MOYENNE(L5C2:L35C2)","L37C2")
=SI(E22=1;ATTEINDRE(A737))
=FORMULE("=MOYENNE(L5C3:L35C3)","L37C3")
=SI(E22=2;ATTEINDRE(A737))
=FORMULE("=MOYENNE(L5C4:L35C4)","L37C4")
=SI(E22=3;ATTEINDRE(A737))
=FORMULE("=MOYENNE(L5C5:L35C5)","L37C5")
=FORMULE("=MOYENNE(L5C6:L35C6)","L37C6")
=FORMULE("=MOYENNE(L5C7:L35C7)","L37C7")
=SELECTIONNER("L4")
=COPIER()
=SELECTIONNER("L36")
=COLLER()
=POSITION(3)
=FORMAT.POLICE("Bookman";12;VRAI;FAUX;FAUX;FAUX;0;FAUX;FAUX)
=SI(OU(A22="performance.mc";A22="performance.mp");ATTEINDRE(A759))
=SELECTIONNER("L4C2:L29C2")
=ENCADREMENT(VRAI;FAUX;;;FAUX)
=SI(E22=1;ATTEINDRE(A774))
=SELECTIONNER("L4C3:L29C3")
=ENCADREMENT(VRAI;FAUX;;;FAUX)
=SI(E22=2;ATTEINDRE(A774))
=SELECTIONNER("L4C4:L29C4")
=ENCADREMENT(VRAI;FAUX;;;FAUX)
=SI(E22=3;ATTEINDRE(A774))
=SELECTIONNER("L4C5:L29C5")
=ENCADREMENT(VRAI;FAUX;FAUX;;;FAUX)
=SELECTIONNER("L4C6:L29C6")
=ENCADREMENT(VRAI;FAUX;FAUX;;;FAUX)
=SELECTIONNER("L4C7:L29C7")
=ENCADREMENT(VRAI;FAUX;;;FAUX)
=SELECTIONNER("L4C2:L36C2")
=ENCADREMENT(VRAI;FAUX;;;FAUX)
=SI(E22=1;ATTEINDRE(A774))
=SELECTIONNER("L4C3:L36C3")
=ENCADREMENT(VRAI;FAUX;;;FAUX)
=SI(E22=2;ATTEINDRE(A774))
=SELECTIONNER("L4C4:L36C4")
=ENCADREMENT(VRAI;FAUX;;;FAUX)
=SI(E22=3;ATTEINDRE(A774))
=SELECTIONNER("L4C5:L36C5")
=ENCADREMENT(VRAI;FAUX;FAUX;;;FAUX)
=SELECTIONNER("L4C6:L36C6")
=ENCADREMENT(VRAI;FAUX;FAUX;;;FAUX)
=SELECTIONNER("L4C7:L36C7")
=ENCADREMENT(VRAI;FAUX;;;FAUX)
=SELECTIONNER("L1C1")
=PLEIN.ECRAN(VRAI)
=FORMULE(0;F23)
=RETOUR()

```

graphe
=EXECUTER(A818)
=SI(F23=1;EXECUTER(A283))

=FORMULE(0;B825)
=AJOUTER.BARRE()
=AJOUTER.MENU(A783;B784:E789)
=AJOUTER.MENU(A783;B791:E796)
=AFFICHER.BARRE(A783)
=SI(OU(A22="performance.mc";A22="performance.mp");ATTEINDRE(A793))
=SI(E22=1;SELECTIONNER("L4C1:L28C2"))
=SI(E22=2;SELECTIONNER("L4C1:L28C3"))
=SI(E22=3;SELECTIONNER("L4C1:L28C4"))
=SI(E22=6;SELECTIONNER("L4C1:L28C7"))
=ATTEINDRE(A797)
=SI(E22=1;SELECTIONNER("L4C1:L35C2"))
=SI(E22=2;SELECTIONNER("L4C1:L35C3"))
=SI(E22=3;SELECTIONNER("L4C1:L35C4"))
=SI(E22=6;SELECTIONNER("L4C1:L35C7"))
=NOUVEAU.DOCUMENT(2)
=SI(B825=1;ATTEINDRE(A808))
=PLEIN.ECRAN(VRAI)
=SELECTIONNER("Axe 2")
=ECHELLE(1;2;1;VRAI;FAUX;FAUX)
=MOTIFS(1;;;3;1;4)
=FORMAT.POLICE(0;1;FAUX;"Bookman";10;FAUX;FAUX;FAUX;FAUX;FAUX;FAUX)
=SELECTIONNER("Axe 1")
=MOTIFS(1;;;3;1;4)
=FORMAT.POLICE(0;1;FAUX;"Bookman";10;FAUX;FAUX;FAUX;FAUX;FAUX;FAUX)
=ATTEINDRE(A810)
=GRAPHIQUE.STANDARD()
=PLEIN.ECRAN(VRAI)
=SI(I780=VRAI;LEGENDE(VRAI);LEGENDE(FAUX))
=SI(I775=1;COURBES(1;VRAI))
=SI(I775=2;HISTOGRAMMES(1;VRAI))
=SI(I775=3;AIRES(1;VRAI))
=SI(I775=4;COMBINAISONS(1))
=SELECTIONNER("")
=RETOUR()
param
=ZONE.DE.DIALOGUE(C773:I782)
=SI(A819=FAUX;ATTEINDRE(A21))
=RETOUR()
rel
=DEFINIR.GRAPHIQUE.STANDARD()
=FERMER(FAUX)
=ACTIVER.SUIV()
=FORMULE(1;B825)
=ATTEINDRE(A787)
sauver
=ENREGISTRER()
=RETOUR()
quitter
=FERMER(VRAI)
=AFFICHER.BARRE(A1)
=RETOUR()
exit
=FERMER(FAUX)
=AFFICHER.BARRE(A1)
=RETOUR()
fleche
=AJOUTER.FLECHE()
=ENTRER("Commentaire sur la flèche :";2;"FLECHE";"Point culminant")
=RETOUR()
titre
=ZONE.DE.DIALOGUE(C847:I850)
=SI(A850=FAUX;RETOUR())
=AJOUTER.TEXTE(1)

```

=FORMULE(ENTRER("TITRE DU GRAPHIQUE :";2;"TITRE";"GRAPHIQUE"))
=FORMAT.POLICE(0;1;FAUX;"Chicago";12;VRAI;FAUX;FAUX;FAUX;FAUX;FAUX)
=MOTIFS(1;;;VRAI;1;;;)
=AJOUTER.TEXTE(3)
=FORMULE(ENTRER("TITRE DE L'AXE DES ABCISSES (JOURS OU MOIS) :";2;"ABCISSES";"Jours"))
=FORMAT.POLICE(0;1;FAUX;"Bookman";10;VRAI;VRAI;FAUX;FAUX;FAUX;FAUX)
=AJOUTER.TEXTE(2)
=FORMULE(ENTRER("TITRE DE L'AXE DES ORDONNEES (NB OU %) :";2;"ORDONNEES";"Nb"))
=FORMAT.POLICE(0;1;FAUX;"Bookman";10;VRAI;VRAI;FAUX;FAUX;FAUX;FAUX)
=SELECTIONNER("")
=RETOUR()

moy
=SI(F23=1;EXECUTER(A282))
=AJOUTER.BARRE()
=AJOUTER.MENU(A867;B784:E789)
=AJOUTER.MENU(A867;B887:E889)
=AFFICHER.BARRE(A867)
=SI(OU(A22="performance.mc";A22="performance.mp");ATTEINDRE(A877))
=SI(E22=1;SELECTIONNER("L29C1:L30C2"))
=SI(E22=2;SELECTIONNER("L29C1:L30C3"))
=SI(E22=3;SELECTIONNER("L29C1:L30C4"))
=SI(E22=6;SELECTIONNER("L29C1:L30C7"))
=ATTEINDRE(A881)
=SI(E22=1;SELECTIONNER("L37C1:L36C2"))
=SI(E22=2;SELECTIONNER("L37C1:L36C3"))
=SI(E22=3;SELECTIONNER("L37C1:L36C4"))
=SI(E22=6;SELECTIONNER("L37C1:L36C7"))
=NOUVEAU.DOCUMENT(2)
=SECTEURS(6;VRAI)
=PLEIN.ECRAN(VRAI)
=LEGENDE(VRAI)
=SELECTIONNER("")
=RETOUR()

titre2
=ZONE.DE.DIALOGUE(C847:I850)
=SI(A889=FAUX;RETOUR())
=AJOUTER.TEXTE(1)
=FORMULE(ENTRER("TITRE DU GRAPHIQUE :";2;"TITRE";""))
=FORMAT.POLICE(0;1;FAUX;"Chicago";12;VRAI;FAUX;FAUX;FAUX;FAUX;FAUX)
=MOTIFS(1;;;VRAI;1;;;)
=SELECTIONNER("")
=RETOUR()

```


[illegible]

[illegible]

CPU		
cpu		
1		
1		
FAUX		
VRAI		
1		

[illegible]

ANNEXE N LE FICHER PERF.INTERPRET DE DEC

Things to look for:

Is the CPU busy? - Use `vmstat(1)` to look at time in cpu states (user, system, idle). I've included an example `vmstat` output to point out the fields (Figure 1). The time in CPU states are the last three fields.

Do you need more memory? - It has been said (of VMS) that ALL performance problems can be solved by adding more memory. This is somewhat true for any VAX and in case of a 4 mg. 11/750 probably a good idea. Use `vmstat(1)` to look at free memory. If it's hovering around 500 free pages on an Ultrix V1.2 system there's a good chance you are paging. Look for free memory in the 5th column (fre).

Are you paging? - Look at columns "re" through "sr". The first two of these (re and at) are soft page faults (don't result in physical I/O. Physical page faults can be watched in "pi" (page in) and "po" (page out).

Is some process I/O bound? - Use `iostat(1)` to look at disk and tty I/O activity. Figure 2 is the output of `iostat(1)` output from a V2.0 system. The V1.x output will have a field for each disk call "msps". I think this is milliseconds per seek. Since DSA type disks aren't command by the OS to seek, it doesn't track this information. If all the I/O activity is showing up on one disk there's a good chance that you have a bottleneck there. I don't have any numbers to tell you what a "busy" disk looks like.

Other things to look at:

What's the load average? - The load average is the average length of the run queue over the last 1, 5 and 15 minutes. Figure 3 is the example output of `uptime(1)`. Depending on the process mix (I/O intensive, CPU intensive, etc) the load average might go down on a faster CPU. The load average doesn't tell you much.

What is the CPU really doing? - Look at the "in", "sy" and "cs" fields of vmstat(1). If the interrupt count is high (in) then look for a device that is very busy. For example UUCP running on a high speed direct connect line to another system will put a High load on the system. A sick device may be interrupting but not doing anything. This will put a load on the system.

If the system call rate (the first sy field) is very high then the system will probably be spending a lot of time in system mode (the 2nd sy field).

The system call rate can be very high. For example:

```
main()
{
    for(;;)
        (void)getpid();
}
```

on my MicroVAX can generate nearly 4000 system calls per second and use about 85% of the system in kernel mode. Sitting idle right now, there are about 15 systems calls per second, 5% user mode, 1% system and rest idle.

Is the network busy? - Figure 4 is a listing of "netstat -i". If there's a network device on the system then you need to watch it. A device like the DEQNA which is getting lots of errors or collisions will put a tremendous load on the system. The DEUNA is a fairly friendly device, but a lot of network traffic could account for the system being busy. The first line of the output is the device counts since the system was booted.

Balanced but slow? - If it looks like you have a good balance* of system usage, but the system is still slow then you might want to pay attetion to the load average. If it seems high** then the machine is probably running at capacity.

Alan

Figure: 1

procs		memory				page				disk				faults		cpu						
r	b	w	avm	fre	re	at	pi	po	fr	de	sr	r0	r1	r2	r3	in	sy	cs	us	sy	id	
1	0	0	944	4546	0	0	0	0	0	0	0	0	1	0	0	0	62	12	4	3	3	94
1	0	0	1576	4500	1	0	6	0	0	0	0	0	5	2	0	0	75	48	18	7	10	83
0	0	0	1618	4404	2	0	8	0	0	0	0	0	6	2	0	0	72	44	20	7	10	84
0	0	0	1523	4366	0	0	2	0	0	0	0	0	3	1	0	0	70	52	20	5	9	86
0	0	0	1222	4354	0	0	0	0	0	0	0	0	0	0	0	0	62	20	7	1	2	97

Figure: 2

tty		ra0		ra1		ra2		ra3				cpu	
tin	tout	bps	tps	bps	tps	bps	tps	bps	tps	us	ni	sy	id
0	10	4	1	1	0	0	0	0	0	3	0	3	94
0	0	0	0	0	0	0	0	0	0	1	0	2	98
1	10	0	0	11	1	0	0	0	0	9	0	4	86
0	0	4	1	0	0	0	0	0	0	1	0	1	98
0	0	0	0	0	0	0	0	0	0	0	0	1	98

Figure: 3

8:49am up 15:28, 1 users, load average: 0.39, 0.21, 0.18

Figure: 4

input (qe0)		output		input (Total)		output	
packets	errs	packets	errs	packets	errs	packets	errs
22673	827	19486	0	367	22867	827	19680
1	0	1	0	0	1	0	1
1	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1

* System balance - I would consider a balanced system to be spending most of its time in user mode (40-60%), with some idle or "nice" (the "ni" field in Figure 2) and the rest system. The disk I/O would be spread evenly across the disk and the ready light (RA type disk) should spend most of it's time on. There will be free memory (> 500) and the pageing columns in vmstat(1) will tend to low or zero. The interrupt, system call and context switch rate should be pretty even. If one is very much higher than the others That could be the problem.

** High Load average - I've been told that a load average of 12 on a 11/780 was high. I know that 6 on my MicroVAX can cause it to be noticeably slower. I don't remember what (decvax) was like when it was a 11/750 (but it was probably over 20). 2 on an 11/730 is busy and anything much higher is capacity.